TSB-UAD: An End-to-End Benchmark Suite for Univariate Time-Series Anomaly Detection

John Paparrizos
University of Chicago
jopa@uchicago.edu

Yuhao Kang
University of Chicago
yuhaok@uchicago.edu

Paul Boniol
Université de Paris
paul.boniol@etu.u-paris.fr

Ruey S. Tsay
University of Chicago
ruey.tsay@chicagobooth.edu

Themis Palpanas
Université de Paris & IUF
themis@mi.parisdescartes.fr

Michael J. Franklin
University of Chicago
mjfranklin@uchicago.edu

ABSTRACT

The detection of anomalies in time series has gained ample academic and industrial attention. However, no comprehensive benchmark exists to evaluate time-series anomaly detection methods. It is common to use (i) proprietary or synthetic data, often biased to support particular claims; or (ii) a limited collection of publicly available datasets. Consequently, we often observe methods performing exceptionally well in one dataset but surprisingly poorly in another, creating an illusion of progress. To address the issues above, we thoroughly studied over one hundred papers to identify, collect, process, and systematically format datasets proposed in the past decades. We summarize our effort in TSB-UAD, a new benchmark to ease the evaluation of univariate time-series anomaly detection methods. Overall, TSB-UAD contains 13766 time series with labeled anomalies spanning different domains with high variability of anomaly types, ratios, and sizes. TSB-UAD includes 18 previously proposed datasets containing 1980 time series and we contribute two collections of datasets. Specifically, we generate 958 time series using a principled methodology for transforming 126 time-series classification datasets into time series with labeled anomalies. In addition, we present data transformations with which we introduce new anomalies, resulting in 10828 time series with varying complexity for anomaly detection. Finally, we evaluate 12 representative methods demonstrating that TSB-UAD is a robust resource for assessing anomaly detection methods. We make our data and code available at www.timeseries.org/TSB-UAD. TSB-UAD provides a valuable, reproducible, and frequently updated resource to establish a leaderboard of univariate time-series anomaly detection methods.

PVLDB Reference Format:
John Paparrizos, Yuhao Kang, Paul Boniol, Ruey S. Tsay, Themis Palpanas, and Michael J. Franklin. TSB-UAD: An End-to-End Benchmark Suite for Univariate Time-Series Anomaly Detection. PVLDB, 15(8): 1697 - 1711, 2022.
doi:10.14778/3529337.3529354

1 INTRODUCTION

A wide range of technological advances in sensing solutions enables collecting enormous amounts of time-varying measurements commonly referred to as *time series*. In particular, analysts estimate that, shortly, billions of Internet-of-Things (IoT) devices will be responsible for generating zettabytes (ZB) of time series [44, 51]. This rapid growth of cost-effective IoT deployments already empowers diverse data science applications and has revolutionized the retail, healthcare, manufacturing, transportation, agriculture, utilities, and automobile industries [80]. Among analytical tasks for IoT data [55, 56, 65, 90], time-series *anomaly detection* is particularly important for identifying abnormal phenomena (either in the behavior of the monitored process, or measurement errors) [8, 49, 54, 82].

Despite over six decades of academic and industrial attention in time-series anomaly detection (AD) [41, 81, 107], only a few efforts have focused on establishing standard means of evaluating existing solutions (notable examples [36, 60, 103, 109, 114, 118]). Unfortunately, there is currently no consensus on using a single benchmark for assessing the performance of time-series AD methods. As a result, we observe two standard practices in the literature for benchmarking AD models by using (i) proprietary and synthetic data; or (ii) a limited collection of publicly available datasets. However, both of these practices are often flawed. In the former case, proprietary or synthetic data may have been collected or generated biasedly to support particular claims, anomaly types, or methods. In the latter case, only a small fraction of datasets are available, some of which suffer from several drawbacks (e.g., trivial anomalies, unrealistic anomaly density, or mislabeled ground truth [114]).

In addition, the ambiguity and the startlingly different interpretation of anomalies across applications further hinders progress. It is not uncommon for methods to achieve high accuracy for some datasets but surprisingly low accuracy for others. The lack of an established benchmark creates the illusion of progress while the identification of robust approaches becomes unlikely. Notably, the recent advances in deep learning technologies have sparked a surge of interest in applying neural network architectures for time-series tasks [37, 38, 40, 92], including for AD [22, 28, 62, 74, 96]. This sudden enthusiasm and a slew of proposed methods in the preceding years underscore the vital need for a time-series AD benchmark.

To address the aforementioned issues and provide an objective means of quantifying the performance of univariate time-series AD methods, we introduce TSB-UAD, an open end-to-end benchmark suite. TSB-UAD serves two purposes. First, to relieve the community from the laborious tasks of identifying, collecting, processing, and formatting relevant datasets that are periodically becoming available. Second, to ease the experimentation through an end-to-end suite for handling pre-processing and post-processing steps, such as data loading, processing, augmentation, transformation, and model evaluation. TSB-UAD performs a rigorous statistical analysis of the results to establish a leaderboard of robust time-series AD methods.

TSB-UAD is the summary of a long process of thoroughly studying over one hundred papers that appeared in the literature in the

Proceedings of the VLDB Endowment, Vol. 15, No. 8 ISSN 2150-8097.
doi:10.14778/3529337.3529354

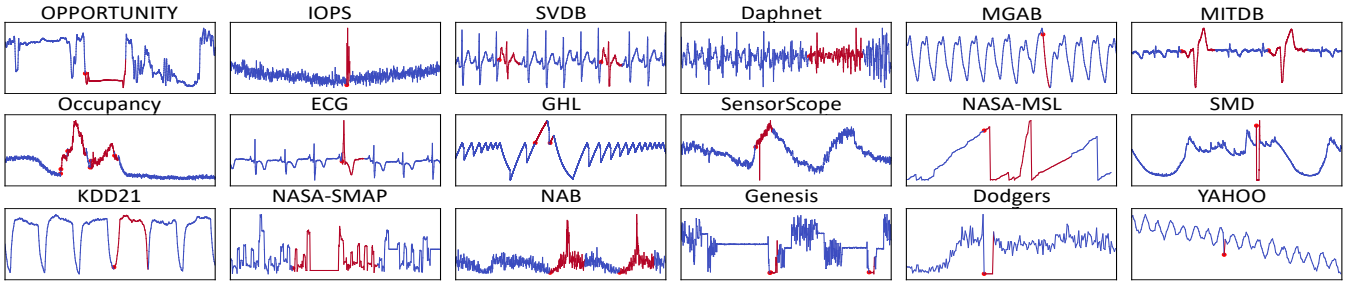


Figure 1: Representative examples of a sample of the public, highly diverse, datasets included in TSB-UAD. The ground truth anomalies are annotated with *red* color. The datasets have high variability in sizes as well as anomaly types, densities, and lengths.

past decades (an extensive collection of the works we considered appear in several recent surveys [13, 23, 49]). Overall, TSB-UAD contains 13766 univariate time series with labeled anomalies from a diverse set of domains and real-world applications that vary significantly in terms of anomaly types, ratios, and sizes. TSB-UAD consists of three dataset categories. The *public* category contains 18 previously proposed datasets with 1980 time series. Figure 1 presents representative examples of a sample of the datasets along with their marked anomalies. Motivated by certain flaws in datasets and evaluation strategies (details in Section 3), we study anomaly types and data transformations to contribute two new collections of datasets. Specifically, the *artificial* category includes 126 datasets with 958 time series. We generate these time series with a principled methodology for transforming time-series classification datasets into time series with labeled anomalies. This methodology relies on a parameter-free classifier to compute the affinity (confusion) matrix among class label, which enables splitting the labels into normal and abnormal. Then, it concatenates sampled time series from these classes to form long time series with a controlled number of anomalies. Despite the artificial generation process, over 90% of these time series correspond to real-world classification time-series data. Finally, considering efforts for developing synthetic time series [36, 60], we study a set of global, local, and subsequence data transformations and produce 92 datasets, using the public datasets, with 10828 *synthetic* time series. Through these transformations, we introduce new outliers with varying difficulty for AD. Importantly, we also study factors that affect the identification of anomalies and propose a set of measures to assess the dataset difficulty.

We believe that TSB-UAD provides an objective means of quantifying the performance of time-series AD methods. The embodied underlying datasets collectively capture significant previous efforts of the past decades, diverse methodologies, and large variability in the characteristics of the anomalies. To state that differently, we can have higher confidence that solutions in the first ranks of TSB-UAD are robust and could likely lead to good performance when applied in a new context. To ease experimentation and ensure reproducibility of our results, we make our data and code available at www.timeseries.org/TSB-UAD. Over time, and with input from the community, we plan to frequently update TSB-UAD datasets and evaluated methods to establish a trusted leaderboard of state-of-the-art time-series AD techniques.

Along with the benchmark suite, we also present an experimental evaluation of 12 representative state-of-the-art AD methods. Our goal is to provide an initial set of recent, strong baselines and illustrate that TSB-UAD is a reliable and robust resource for evaluating time-series AD methods. Our findings corroborate our claim and demonstrate the difficulty of methods to consistently perform well across such a diverse set of time series and anomaly types. TSB-UAD's rigorous statistical methodology enables analysis over different levels of granularity (i.e., aggregated analysis per dataset or fine-grained analysis per time series) and across different types and densities of anomalies, which reveals new insights about the performance of AD methods. For example, modern deep learning methods perform exceptionally well for point-based anomalies but poorly for subsequence-based anomalies. In other cases, surprisingly simple methods outperform complex solutions. By introducing new anomalies and by varying parameters of data transformations, TSB-UAD can assess the robustness of methods under different scenarios and varying difficulty, revealing cases where the performance of methods can substantially alter. Our results demonstrate the usefulness of TSB-UAD for evaluating methods for time-series AD.

We start with a discussion of the related work for time-series AD (Section 2) and we review several disparities in the benchmarking of AD methods (Section 3). Then, we present our contributions:

- We review over one hundred papers in the literature to identify, collect, process, and bring in a unified format 18 previously proposed datasets for univariate time-series AD (Section 4.1).
- We describe a principled methodology for generating labeled AD datasets from time-series classification datasets in order to leverage decades of effort in that area (Section 4.2).
- We study data transformations to assist in the augmentation of datasets with new, more complex anomalies (Section 4.3).
- We report the evaluation measures and the rigorous statistical analysis included in TSB-UAD (Section 4.4).
- We review factors affecting the performance of methods and introduce measures to assess the dataset difficulty (Section 4.5).
- We present an initial experimental evaluation of recent representative methods on TSB-UAD (Sections 5 and 6).

Finally, we conclude with the implications of our work (Section 7).

2 RELATED WORK

Anomaly Detection: The importance of AD in time-series data was recognized well before the inception of computer science [81]. In 1972, Fox conducted the first study to examine anomalous behavior across time and defined two types of outliers [41]. In 1988, Tsay extended these outliers into four types for univariate time series [107] and subsequently for multivariate time series [109]. Due to the large variety of applications, domains, and anomaly types, every year, a vast number of papers appear in the literature proposing new methods for AD in time series, and it is beyond our scope to cover extensively here. Next, we will only briefly summarize popular categories of methods, and we refer the reader to three recent survey papers for a detailed coverage of such methods [13, 23, 49].

Discord-based methods focus in the analysis of subsequences for the purpose of detecting anomalies in time series, mainly by utilizing nearest neighbor distances among subsequences [26, 43, 58, 64, 68, 99, 117]. Instead of measuring nearest neighbor distances, *proximity-based methods* focus on detecting globally normal or isolated behaviors. General-purpose multi-dimensional point outlier methods have been proposed in this category [24, 66, 69], with Isolation Forest [66] working particularly well when extended for subsequences [18]. Recent methods in this category first cluster data to obtain the normal behavior and seem to achieve competitive performance [15–21]. To provide ordering information for subsequences, methods may leverage graph representations in which edges encode the ordering [18]. Alternative methods exist for anomaly detection but they are not specifically designed for subsequence AD [12, 59, 83, 102, 106]. Finally, deep learning approaches, e.g., based on recurrent [73] or convolutional [77] neural networks, have been proposed for this task. We refer the reader to a recent survey for a detailed coverage of deep learning methods for AD [28].

Benchmarks: Our community has a long tradition in publishing benchmarks for evaluating the performance from traditional database systems [47, 78], big data systems [45], key-value stores [30], stateful services [14], and streaming systems [4], to, more recently, machine learning and deep learning applications [29, 94]. Similarly, the machine learning community has also devoted substantial effort publishing datasets and benchmarks for many applications [9, 33, 111]. For time series, in particular, there are already several resources containing datasets useful for time-series forecasting, classification, and clustering tasks [6, 31, 34, 70–72]. Unfortunately, despite the evident need for datasets for AD, much less work has been devoted in that direction. Specifically, the majority of published work rely either on generators for synthetic data (e.g., [60]) or on limited available dataset (e.g., NAB [3] and Yahoo [61]). Several drawbacks in these datasets have resulted in recent criticism [100, 114]. Finally, regarding evaluation, the traditional *Precision* and *Recall* measures have been extended to consider ranges and enable adequate evaluation for time-series AD [104]. Existing benchmarks, such as the Exathlon benchmark [53] and the KDD21 competition [114] focus on anomalies on a single application and a single anomaly type, respectively. In contrast, TSB-UAD comprehensively covers a diverse set of domains, applications, anomaly types, and data transformations to generate anomalies of increasing difficulty. TSB-UAD aims to ease the evaluation by enabling integration of new detectors while automating the remaining tasks.

3 BIASES IN TIME-SERIES AD EVALUATIONS

The lack of an established benchmark for time-series AD often leads to biases in the selection of datasets, parameters, and evaluation measures. Unfortunately, these biases introduce disparities in the experimentation of published works. We start by describing the different time-series anomaly types (Section 3.1). Then, we discuss three core disparities in existing evaluation strategies after thoroughly studying the literature (Sections 3.2, 3.3, and 3.4). Finally, we overview criticism on certain flaws of the available AD datasets (Section 3.5), supporting the need to establish a new comprehensive benchmark. We avoid references as our goal is to raise awareness of these pervasive issues and not to criticize specific works.

3.1 Types of Time-Series Anomalies

There are three types of time-series anomalies: *point*, *contextual*, and *collective* anomalies. *Point* anomalies refer to data points that deviate remarkably from the rest of the data. Figure 2(a) depicts a synthetic time series with a point anomaly: the value of the anomaly is outside the expected range of normal values. *Contextual* anomalies refer to data points within the expected range of the distribution (in contrast to point anomalies) but deviate from the expected data distribution, given a specific context (e.g., a window). Figure 2(b) illustrates a time series with a contextual anomaly: the anomaly is within the usual range of values (left distribution plot of Figure 2(b)) but outside the normal range of values for a local window (right distribution plot of Figure 2(b)). *Collective* anomalies refer to sequences of points that do not repeat a typical (previously observed) pattern. Figure 2(c) depicts a synthetic collective anomaly. Figure 1 contains representative examples of all three anomaly types. The first two categories, namely, point and contextual anomalies, are referred to as *point-based* anomalies. whereas, *collective* anomalies are referred to as *sequence-based* anomalies.

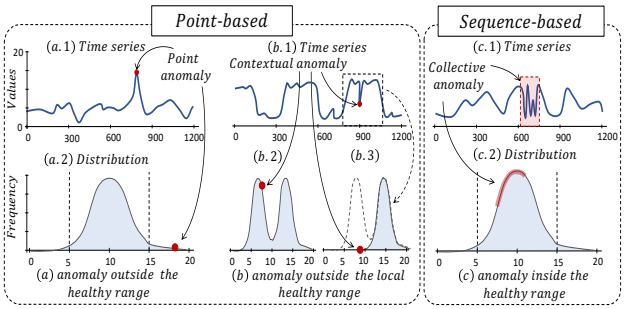


Figure 2: Synthetic illustration of the three time-series anomaly types: (a) point; (b) contextual; and (c) collective anomalies.

3.2 Selection of Datasets

The selection of datasets can significantly influence the experimental outcome. Despite the ubiquitous understanding that there are different types of time-series anomalies [13, 23, 49], evaluating proposed methods only on a limited set of datasets (often synthetic or proprietary) with only one such anomaly type is common. We believe that to have a rational assessment of a technique, a benchmark should include all types of datasets and anomalies. This trait is standard in established benchmarks for different time-series tasks, such as for classification (UCR archive) [31] or forecasting (M competitions) [70–72]. For example, the UCR archive includes over 100 datasets spanning a diverse set of domains with wide variability in terms of size, length, number of classes, and class imbalance. A benchmark with such variability for AD can help understand trade-offs between general-purpose or anomaly-specific methods.

Excluding datasets from an evaluation poses a substantial barrier to consistently analyzing different algorithms across the three previously defined categories of anomalies. For example, point-based forecasting methods, which detect anomalies by comparing predictions to actual data, may perform well in Yahoo (with point anomalies) and poorly in NAB (with collective anomalies). Thus, focusing solely on Yahoo, which contains over 300 time series, may seem as an extensive evaluation, yet, the method's evaluation is skewed due to the incompleteness of the benchmark. Newcomer researchers and practitioners who are in dire need of assessing their own proposed models (as well as reviewers evaluating a scientific work), may be oblivious of such issue and accept the outcome (e.g., it is significant to outperform methods in 300 time series, even though one-line-of-code baselines may achieve comparable results

to complex methods [114]). Importantly, justifying the dataset selection based on a particular application or anomaly category is unrealistic. For example, ECG data may mainly contain collective anomalies (and not individual points). However, there is no guarantee that in a realistic setting, a user malfunction or an imperfection in the measurement system cannot generate other anomaly types.

3.3 Selection of Model Parameters

Beyond selecting datasets for covering a diverse set of anomaly types, similarly significant are the intrinsic characteristics of the datasets, which affect the choice of model parameters during evaluation. Among standard datasets, some contain many anomalies of varying duration and high anomaly contamination (e.g., the NASA-SMAP [11]) whereas others (e.g., NAB [3]) limit the anomalies to a few of specific duration. Other datasets (e.g., KDD21 [57]) limit the anomalies per dataset to one and evaluate models using a single index instead of the entire length of the anomaly.

Such disparities in the formatting of anomalies and the characteristics of datasets render it challenging to adjust necessary model parameters. As a simple example, we can consider the variety of ways in which existing methods report anomalies. Specifically, some methods return a raw anomaly score per point, requiring the operator to manually establish a threshold value to extract anomalies. Other methods simplify the process by adapting the threshold to the score. Similarly, some methods return anomalies with a fixed length while others of variable length. Therefore, only tuning parameters relevant to the scoring of anomalies becomes cumbersome. The parameter selection process becomes chaotic for methods with many additional parameters (e.g., deep learning approaches). By analyzing dozens of papers and their code repositories, it was evident that, in some instances, there was an inadequate tuning of parameters or cases of overfitting. An open benchmark requiring evaluations across datasets with established, reproducible results can eliminate the aforementioned issues.

However, parameter choices may also offer disproportionate benefits to particular methods, as we will see (Section 6). For example, the KDD21 datasets may contain more than one anomaly, but their ground truth data focuses on the most "prominent" anomaly. Therefore, methods, which by design extract a single anomaly, may perform well under such a setting but their performance may degrade in settings with high contamination of anomalies. Similarly, methods extracting all anomalies but not ranking them appropriately may get penalized. High variability of characteristics is desirable to assess both the effectiveness and the efficiency of methods.

3.4 Selection of Evaluation Measures

The choice of measure to quantify the quality of methods may also significantly bias the experimental outcome. A wide range of measures has been used to evaluate AD methods. Briefly, traditional measures, such as Precision, Recall, and F-score, assess the methods by assuming each time-series point can be marked as an anomaly or not (e.g., by a threshold on an anomaly score). Shortcomings of these measures (i.e., difficulty in evaluating collective anomalies) motivated the creation of range-based variants [104]. Therefore, selecting datasets with collective anomalies while using the traditional measures may result in misleading outcomes. Interestingly, previously mentioned measures require setting a threshold to mark points as anomalies or not. The AUC measure, on the other side, eliminates such a need as its value is independent of a threshold. Returning to the above example of KDD21, which contains multiple anomalies but only the top-1 appears in ground truth, the selection of AUC could avoid partially biases because it avoids setting a threshold on the anomaly score for extracting a single anomaly.

3.5 Flaws in Limited Available Datasets

Apart from the biases that arise from selecting datasets, parameters, and evaluation measures, flaws in current datasets may also result in misleading outcomes. In particular, certain available time-series AD datasets suffer from several drawbacks, resulting in criticism [114] for works focusing solely on them for their evaluation.

In summary (see details in [114]), for some datasets, a trivial solution, defined simply as one line of code "baseline" using standard functions (e.g., mean, std, etc.) and some tuned parameters, may achieve state-of-the-art performance. In other cases, datasets contain a high density of anomalies tailoring the problem more into classification. For datasets where most anomalies appear in the end, they may provide opportunities for biasing predictions towards the last points. Finally, mislabeling issues, where only some anomalies are marked, may lead to false positives and negatives.

Notably, several of the dataset flaws are problematic mainly due to cherry-picking or the disparities in selecting parameters or evaluation measures, as mentioned earlier. For example, a simple method performing well (without brute-force parameter search) across *all* such "trivial" datasets is still valuable. Stating that differently, if a technique performs well on some more challenging datasets, it would be worrisome to not perform well in such "trivial" datasets. In addition, high anomaly density or anomalies appearing towards the end become less relevant if the evaluation focuses on a large variety of datasets (e.g., a method predicting anomalies in the last points will no longer succeed across all datasets).

Nevertheless, there is a clear need for the community to agree on benchmarks that are accurate (i.e., with no wrong labels), diverse (i.e., originating from different domains, having different data characteristics, including different types of anomalies), and that properly test all different aspects of the relevant algorithms.

4 TSB-UAD: BENCHMARK DETAILS

Given the above disparities and the inconsistencies in some datasets, we studied many papers appearing in the literature in the past two decades [13, 23, 49] to identify the best practices in time-series AD benchmarking and collect previously used datasets. We summarize this lengthy process by introducing TSB-UAD, our end-to-end time-series AD benchmarking suite. First, we review the categories of datasets and the library included in TSB-UAD (Section 4.1). Second, we describe a principled methodology for generating AD datasets from time-series classification datasets to leverage decades of efforts in that area (Section 4.2). Third, we study global, local, and subsequence data transformation to assist in the augmentation of datasets with new, more complex anomalies (Section 4.3). Then, we report the evaluation measures and the rigorous statistical analysis included in TSB-UAD (Section 4.4). Finally, we study factors affecting the performance of methods and introduce measures to assess the dataset difficulty for time-series AD (Section 4.5).

4.1 Datasets and Benchmarking Suite

TSB-UAD consists of three dataset categories, namely, public, artificial, and synthetic datasets. *Public* datasets contain previously proposed datasets appearing in the literature across different communities. *Artificial* datasets include mainly real-world datasets (over 90%) used previously for time-series classification and transformed into AD datasets with labeled anomalies. *Synthetic* datasets are

Table 1: Summary characteristics of the 18 public datasets included in TSB-UAD. R_c is the relative contrast (discussed in Section 4.5), a coefficient measuring the distribution of normal and abnormal points, with smaller values indicating relatively higher difficulty.

Dataset	Count	Average Length	Average # Anomalies	Average # Abnormal Points	R_c
Dodgers [52]	1	50400.0	133.0	5612.0	2.02
ECG [75]	52	230351.9	195.6	15634.0	8.33
IOPS [1]	58	102119.2	46.5	2312.3	3.33
KDD21 [57]	250	77415.06	1	196.5	10.67
MGAB [105]	10	100000.0	10.0	200.0	27.64
NAB [3]	58	6301.7	2.0	575.5	2.67
SensorScope [115]	23	27038.4	11.2	6110.4	2.38
YAHOO [61]	367	1561.2	5.9	10.7	3.25
NASA-MSL [11]	27	2730.7	1.33	286.3	1.97
NASA-SMAP [11]	54	8066.0	1.26	1032.4	4.18
Daphnet [5]	45	21760.0	7.6	2841.0	2.38
GHL [39]	126	200001.0	1.2	388.8	27.24
Genesis [110]	6	16220.0	3.0	50.0	2.28
MITDB [76]	32	650000.0	210.1	72334.3	7.19
OPP [95]	465	31616.9	2.0	1267.3	2.94
Occupancy [27]	10	5725.8	18.3	1414.5	2.53
SMD [101]	281	25562.3	10.4	900.2	3.39
SVDB [48]	115	230400.0	208.0	27144.5	7.14

augmented versions of the public datasets where various data transformations infuse new anomalies or increase their complexity.

Public Datasets: We identified and collected 18 datasets proposed in the past decades in the literature containing 1980 time series with labeled anomalies. Specifically, each point in every time series is labeled as normal or abnormal. Table 1 summarizes relevant characteristics of the datasets, including their size and length, as well as statistics about the anomalies. The first 8 datasets originally contained univariate time series, whereas the remaining 10 datasets originally contained multivariate time series that we converted into univariate time series. Specifically, we run our AD methods (Section 6) on each dimension separately, and we keep those dimensions where at least one method achieves $AUC > 0.8$ (Section 4.4).

Even though some of these datasets are publicly available (e.g., in code repositories), we could not identify works performing evaluations in a large portion of them. The main reason is the laborious task of identifying and collecting datasets across different communities and, subsequently, processing and formatting the datasets to bring them in a unified format. For some datasets, complicated documentation describes the collection process and instructions for extracting anomalies. In other cases, the lack of documentation hinders the process of utilizing the datasets. We relieve the community from this task and provide datasets in a unified format with the scripts for extracting anomalies from the original data.

Briefly, TSB-UAD includes the following datasets:

- Dodgers [52] is a loop sensor data for the Glendale on-ramp for the 101 North freeway in Los Angeles and the anomalies represent unusual traffic after a Dodgers game.
- ECG [75] is a standard electrocardiogram dataset and the anomalies represent ventricular premature contractions. We split one long series (MBA_ECG14046) with length ~ 1e7) to 47 series by first identifying the periodicity of the signal.
- IOPS [1] is a dataset with performance indicators that reflect the scale, quality of web services, and health status of a machine.
- KDD21 [57] is a composite dataset released in a recent SIGKDD 2021 competition with 250 time series.
- MGAB [105] is composed of Mackey-Glass time series with non-trivial anomalies. Mackey-Glass time series exhibit chaotic behavior that is difficult for the human eye to distinguish.
- NAB [3] is composed of labeled real-world and artificial time series including AWS server metrics, online advertisement clicking rates, real time traffic data, and a collection of Twitter mentions of large publicly-traded companies.
- NASA-SMAP and NASA-MSL [11] are two real spacecraft telemetry data with anomalies from Soil Moisture Active Passive (SMAP) satellite and Curiosity Rover on Mars (MSL). We only keep the first data dimension that presents the continuous data, and we omit the remaining dimensions with binary data.
- SensorScope [115] is a collection of environmental data, such as temperature, humidity, and solar radiation, collected from a typical tiered sensor measurement system.
- Yahoo [61] is a dataset published by Yahoo labs consisting of real and synthetic time series based on the real production traffic to some of the Yahoo production systems.
- Daphnet [5] contains the annotated readings of 3 acceleration sensors at the hip and leg of Parkinson's disease patients that experience freezing of gait (FoG) during walking tasks.
- GHL [39] is a Gasoil Heating Loop Dataset and contains the status of 3 reservoirs such as the temperature and level. Anomalies indicate changes in max temperature or pump frequency.
- Genesis [110] is a portable pick-and-place demonstrator which uses an air tank to supply all the gripping and storage units.
- MITDB [76] contains 48 half-hour excerpts of two-channel ambulatory ECG recordings, obtained from 47 subjects studied by the BIH Arrhythmia Laboratory between 1975 and 1979.
- OPPORTUNITY [95] (OPP) is a dataset devised to benchmark human activity recognition algorithms (e.g., classification, automatic data segmentation, sensor fusion, and feature extraction). The dataset comprises the readings of motion sensors recorded while users executed typical daily activities.
- Occupancy [27] contains experimental data used for binary classification (room occupancy) from temperature, humidity, light, and CO2. Ground-truth occupancy was obtained from time stamped pictures that were taken every minute.
- SMD [101] (Server Machine Dataset) is a 5-week-long dataset collected from a large Internet company. This dataset contains 3 groups of entities from 28 different machines.
- SVDB [48] includes 78 half-hour ECG recordings chosen to supplement the examples of supraventricular arrhythmias in the MIT-BIH Arrhythmia Database.

Artificial Datasets: To leverage existing real-world datasets used for alternative time-series tasks, we followed established work for systematic construction of AD datasets from generic classification datasets [36]. We propose a version suitable for time series that splits class labels into normal and abnormal (see Section 4.2). This process enables the generation of 958 time series, belonging to 126 datasets corresponding to datasets of the UCR Archive [31]. Despite this artificial generation process, these time series correspond mainly to real-world datasets and applications (over 90%).

Synthetic Datasets: Considering significant efforts for developing synthetic time series for AD [60], we study and present a set of global, local, and subsequence data transformations with the purpose to infuse new anomalies or increase the complexity of identifying existing anomalies (see Section 4.3). We apply these transformations in the public datasets to produce 92 augmented datasets, with 10828 time series. Importantly, we provide the corresponding scripts to assist in the creation of exponentially large datasets with varying difficulty to evaluate time-series AD methods.

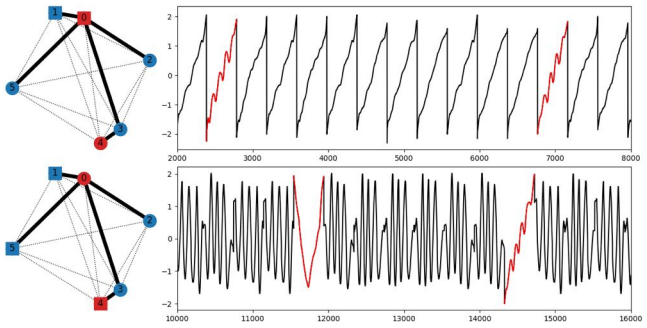


Figure 3: Artificial AD dataset generation process from a time-series classification dataset (Symbols from [31]). In the left panel, the solid lines indicate the MST based on the affinity matrix among all labels. The neighboring nodes belong to two different colors, red (blue) label is abnormal (normal). Square labels are selected to generate the time series. In the right panel, we present two sample time series.

TSB-UAD Benchmarking Suite: The aim of the accompanied Python library is to hide all the complexity of benchmarking AD methods by handling pre-processing and post-processing steps (that we discuss in next sections), such as data loading, processing, generation, and transformation, model evaluation, and rigorous statistical analysis. Researchers and practitioners should mainly focus their attention on implementing *detectors*, the methods for extracting anomalies. In our repository, we provide 12 examples of supervised and unsupervised detectors that we use in our evaluation. We also release all the datasets across all categories for reproducibility purposes and for utilization outside of the accompanied library.

4.2 Transforming Time-Series Classification Datasets into Labeled AD Datasets

A principled methodology is necessary to leverage existing time-series classification datasets by transforming the class labels assigned to each time series into normal and abnormal. This is of great importance considering decades of effort behind the creation of such classification datasets. If we can achieve a distinction between normal and abnormal classes, we can then produce time-series AD datasets by sampling (and concatenating) time series from the normal classes while controlling the anomaly density during the infusion of time series sampled from the abnormal classes.

A key challenge is to avoid the creation of time series with trivial or impossible to detect anomalies. To achieve this goal, we proceed in two steps (following the high-level framework in [36]): (1) we identify pairs of classes with the high confusion, given by an oracle classifier; and (2) for selected class pairs, we generate time series but retain only cases where at least one detector can identify the anomalies. By using different thresholds to assess the detection accuracy (see Section 4.4), we can generate datasets of varying difficulty. For TSB-UAD, we construct the dissimilarity matrix for time series using the SBD distance [10, 88, 89]. SBD is a fast, accurate, and parameter-free distance measure that has achieved state-of-the-art performance [91] and is suitable to incorporate the temporal structure of time series. Then we apply a 1-NN (Nearest Neighbor) classifier, using the pre-computed distance matrix, and predict the label of each sequence. Even though more accurate classifiers exist [7, 35, 84, 87], we favor a deterministic and parameter-free classifier to ensure reproducible results.

Table 2: Summary characteristics of 10 datasets (out of 126) in artificial category of TSB-UAD. R_c is the relative contrast (see Section 4.5), a coefficient measuring the distribution of normal and abnormal points, with smaller values indicating relatively higher difficulty.

Dataset	Count	Average Length	Average # Anomalies	Average # Abnormal Points	R_c
Earthquakes	8	58624	2	1024	1.67
Semg...Ch2	2	76750	2	1000	2.90
ElectricDevices	6	9984	2	192	4.10
UWave...Z	2	32445	2	630	5.51
Worms	8	91800	2	1800	5.89
Distal...Correct	2	8160	2	160	6.37
AllGesture...Z	5	51000	2	1000	6.72
AllGesture...X	8	57625	2	1000	9.38
ECG5000	7	16280	2	280	10.95
EOG...Signal	8	57250	2	1000	13.47

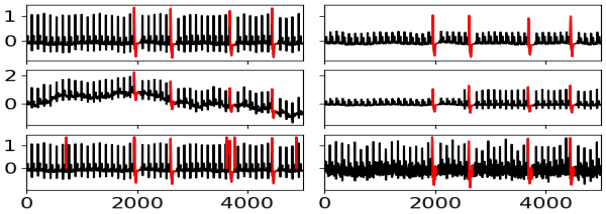
The probability that a series x with label $y = j$ is predicted to be label $\hat{y} = k$, $P(\hat{y} = k|x)$, is defined as the confusion factor c_{jk} from label j to label k. The normalized affinity between label j and k is $(c_{jk} + c_{kj})/2$. We construct the affinity matrix between all m labels and compute the maximum spanning tree (MST) based on this matrix. Alternatively, someone might consider picking as normal and abnormal classes from the single *pair* of classes with the maximum confusion. Unfortunately, this will result in cases very difficult to distinguish. Instead, the MST step helps to construct sets that have the largest overall affinity. Finally, we two-color the adjacent nodes in the MST and assign one color as normal and the other as abnormal. Two parameters are necessary to generate data: the number of abnormal segments K and the anomalous subsequence ratio r. The number of normal segments is $N = K/r - K$. Suppose the abnormal set contains m labels with n data. We pick K data without replacement from n data. If $n < K$, then we pick all n data. We count the frequency f_a of the most frequent anomalous label, then the frequency of each normal label $f_n = 20f_a$. The number of selected normal labels is $L = N/f_n$. We pick L normal labels, and each normal label is selected f_n times with replacement. Finally, K anomalous segments and N normal segments are shuffled and concatenated to form one synthetic time series.

We use this process on all 128 datasets in the UCR archive (after fixing issues with varying lengths and missing values [85]) to generate one or more time series per dataset. We only keep time series for which at least one of five unsupervised methods that performed well in the public datasets achieved AUC higher than 0.65. This process excluded completely two datasets and several generated time series from each dataset, resulting in 126 datasets with 958 time series. Figure 3 presents the MST and the corresponding time series generated along with the infused anomalies (in red) for the Symbols dataset. Table 2, summarizes relevant characteristics of 10 representative datasets in the artificial category of TSB-UAD.

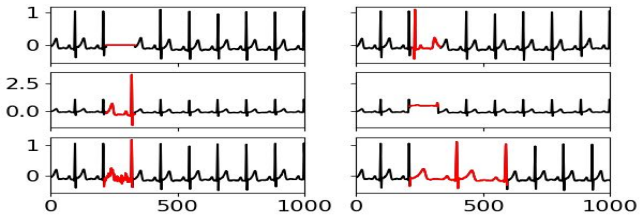
4.3 Data Transformations for Synthetic Dataset Generation of Increasing Difficulty

To emulate additional anomalies or increase the complexity for AD, we study a set of global, local, and subsequence transformations. *Global* transformations change characteristics of the entire time series, while the *local* and *subsequence* transformations modify contiguous portions of the time series. We denote the original time series as $X = (x_0, x_1, ..., x_n)$ with standard deviation s.

4.3.1 Global Transformations: We consider five global transformations to alter time series using random walk background, white



(a) Global transformations. Left: original data (top), random walk background (middle), add point outliers (bottom). Right: smoothing (top), apply two smoothing windows (middle), add white noise (bottom).



(b) Local transformations. Left: insert a flat region (top), re-scale one period (middle), add white noise to a period (bottom). Right: flip one period horizontally (top), tanh normalization (middle), over sampling (bottom).

Figure 4: Illustrations of global and local data transformations.

noise, point outliers, smoothing, and smoothing of segments. Specifically, we define p_{rw} as the strength of a *random walk background*. Let us assume an independent random variable $Z_i, i \in [1, n]$ from [-1,0,1] and set $b_0 = 0$ and $b_n = \sum_{i=1}^{n} Z_i$. We transform the original time series as follows: $x'_i = x_i + p_{rw} \cdot s \cdot b_i$. Similarly, we define p_{wn} as the strength of *white noise* and $b_i \sim \mathcal{N}(0, 1), i \in [0, n]$. We transform the original time series as follows: $x'_i = x_i + p_{wn} \cdot s \cdot b_i$. The integrated backgrounds do not alter the label of each point, however, they increase the AD difficulty for certain methods. For *point outliers*, we define the outlier ratio p_{or} and randomly pick $n \cdot p_{or}$ points from the time series. The selected data point is altered as follows: $x'_i = x_i + 5s$. Another version replaces the selected data with the maximum value of the original time series: $x'_i = max(X)$. Thus, point outliers may introduce global or contextual anomalies. For *smoothing*, f is a Gaussian filter with line width p_{sm}. The smoothed time series is the convolution between original time series X and filter f, which can be calculated through the product of their Fourier transforms, $X' = X * f = \mathcal{F}^{-1}(\mathcal{F}(X)\mathcal{F}(f))$. Finally, we can extend this idea to multiple segments by altering *two segments with different smoothing windows*. We apply two Gaussian filters with different line widths p_1, p_2 to the first and second half of the time series, respectively, and concatenating them together $X' = [X_{0:n/2} * f_{p_1}, X_{n/2:n+1} * f_{p_2}]$.

4.3.2 Local Transformations: We consider two local transformations designed to emulate collective anomalies. *Pattern-related* transformations insert a plateau region; flip a segment horizontally; re-scale or shift a segment with coefficient α; normalize a segment with z-score, MinMax, MedianNorm, MeanNorm, Logistic, and Tanh [91]; or add white noise to one segment. *Frequency-related* transformations re-sample a segment with a new frequency. Similar to the point outlier ratio, we can also define the transformed-period ratio and apply these local transformations to all selected periods. Figure 4 shows the effect of several of the global and local data transformations when applied to a sample time series.

4.3.3 Subsequence Transformations: The third type of transformation is limited to the synthetic datasets. Given that we set the normal and abnormal sequences, we can control the anomalous subsequence ratio, the affinity level among anomalous data, and the affinity level among anomalous and normal data. For instance, Figure 3 displays two synthetic time series based on the same dataset. When we pick label 1 as the anomalous data and label 0 as the normal data (upper panel), the two anomalous subsequences are similar since they belong to the same label. The normal and anomalous subsequences are also similar. When we pick labels 1 and 5 as the anomalous data and label 4 as the normal data (lower panel), then the distance between two anomalous subsequences and the distance between anomalous and normal subsequences are both evident. To generate examples, we follow the method discussed earlier to first color (mark) the classes and we randomly select classes from different colors. The ensembled data will contain various levels of affinity and spurious anomalies (filtered by black-box AD methods).

We note that the initial transformations in TSB-UAD capture a part of the overall spectrum of options. We plan to incorporate more advanced transformations, such as Markov Switching models or varying-parameter models [93, 108], in the near future.

4.4 Evaluation Measures and Tests for AD

Several measures have been proposed to quantify the quality of AD methods. Next, we review these measures and present some of their shortcomings. TSB-UAD includes *all* measures for completeness.

Precision, Recall, and F-score: Let P and N be the number of actual positive and negative points, TP, FP, TN, FN be the number of true positive, false positive, true negative, and false negative classifications. Then we define $Precision = TP/(TP + FP)$, True positive rate (Recall) $TPR = TP/P = TP/(TP + FN)$, and False positive rate $FPR = FP/N = FP/(FP + TN)$. Subsequencely, F-score is the harmonic mean of the precision and recall: $F - score = 2 \cdot Precision \cdot Recall/(Precision + Recall)$. These measures are most commonly used to evaluate AD methods in the literature.

AUC: Precision, Recall, and F-score depend on the need to select a threshold on the anomaly score to determine normal and abnormal points (such that the TP, FP, TN, FN quantities can be computed). A comprehensive evaluation is to vary the threshold from the highest to the lowest possible value, given a computed anomaly score from an AD method. To achieve that, we use the receiver operating characteristics curve (ROC) to record the relationship between the TPR and FPR during this process. The area under the curve (AUC) is an appropriate measure to compare AD method. AUC ignores the precision and, therefore, AUC may be over-optimistic for an unbalanced sample. For such cases, F-score is a good supplement or a Precision-Recall curve (both included in TSB-UAD).

Range-Precision and Range-Recall [104]: To alleviate shortcomings of the traditional Precision and Recall measures, their definitions were extended recently to capture ranges. Specifically, their definition considers several factors: the ratio of the number of detected anomaly subsequences to the total number of the anomaly subsequences, the ratio of the number of the detected point outliers to the total number of the point outliers, the relative position of the true positive portion in each anomaly subsequence, the number of the fragmented prediction regions that correspond to one real anomaly subsequence. As before, the Range-F-score is the harmonic mean of the Range-Precision and the Range-Recall.

Precision@k: Finally, given the number of abnormal points k, an alternative measure expects the points with the k most significant anomaly scores as anomalies. By denoting the corresponding TP as TP@k, then Precision@k = TP@k / k.

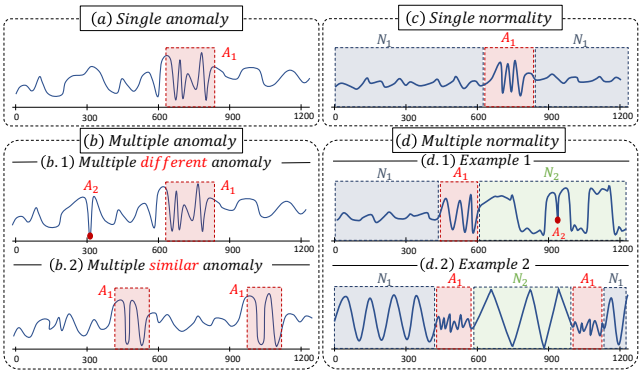


Figure 5: Synthetic illustration of six time series containing: (a) one single anomaly; (b.1) two distinct anomalies; (b.2) two similar anomalies; (c) one single normality; and (d.1/2) two different normalities.

As noted earlier, TSB-UAD reports all measures to provide a complete picture of each method. In our experiments, we evaluate the sensitivity of all measures to several factors (Section 6.1) and we use AUC due its robustness to noise and the normal/abnormal ratio. Importantly, AUC also avoids the dependency of all other measures in setting threshold parameters on the anomaly scores. Moreover, TSB-UAD is a modular benchmark, which enables integration of new measures (e.g., Range-AUC and VUS-based measures [86]).

Statistical Analysis: Considering that we perform evaluations across different datasets, appropriate statistical tests are necessary to assess the significance of the differences in accuracy. For TSB-UAD, we employ two non-parametric statistical tests: one to validate pairwise comparisons and one to validate multiple methods together. Specifically, following [32], we use the Wilcoxon test [112] to evaluate pairs of methods over multiple datasets. To reason about multiple methods over multiple datasets we use the Friedman test [42] followed by the post-hoc Nemenyi test [79].

4.5 Assessing Difficulty of the AD Datasets

In addition to evaluating methods, we also consider factors for assessing the difficulty of the time series for AD. To illustrate this point, Figure 5 presents synthetic time series divided into two blocks. In the first block, Figure 5(a) depicts a time series with only one anomaly, Figure 5(b.1) illustrates a time series with two different anomalies, and Figure 5(b.2) depicts a time series with two similar (in shape) anomalies. In the second block, Figure 5(c) depicts a time series with single normality and Figure 5(d) shows a time series with two different normalities. Based on those two cases, we summarize several factors affecting the detection of anomalies: (1) the variation of normal subsequence: a time series with a single or multiple normal patterns (see Figure 5(c) versus (d)); (2) the variation of anomalies: a time series with a single type of anomaly or multiple different anomalies (see Figure 5(a) versus (b.1)); and (3) the cardinality of anomalies: the time series contain unique or multiple similar anomalies (see Figure 5(b.1) versus (b.2)).

As we will see, different combinations of these factors may result in different appropriate methods for AD. Several coefficients have been previously discussed to capture the distribution of normal and abnormal points [36, 50]. We generalize these definitions for time series and propose a measure suitable for the synthetic datasets.

Relative Contrast (RC) [50]: RC is defined as the ratio of the expectation of the mean distance to the expectation of nearest neighbor distance for all data points. Denote SBD distance between two series s_i and s_j as $D(s_i, s_j)$ and the whole set is S. For each series x, its 1-NN distance $D_{min}(x) = \min_{s \in S-x} D(x, s)$ and mean distance $D_{mean}(x) = \mathbb{E}_{s \in S-x} D(x, s)$. The relative contrast:

$$R_c = \frac{\mathbb{E}_{s \in S}[D_{mean}(s)]}{\mathbb{E}_{s \in S}[D_{min}(s)]}.$$

To calculate RC, we first split the time series by its period, and the distance matrix is built based on the SBD distance among each pair of subsequences. RC measures the separability of the nearest neighbor of a point from the other points. If RC is closer to 1, the mean distance is close to the nearest-neighbor distance, which indicates the data distributes uniformly, and the clustering is harder.

Normalized clusteredness of abnormal points (NC) [36]: NC is the ratio of the average SBD of normal subsequences to the average SBD of abnormal subsequences. Denote the set of normal sequences S_{nor} and the set of anomalous sequence S_{ano}, then

$$N_c = \frac{\mathbb{E}_{s_i, s_j \in S_{nor}, i \neq j}[D(s_i, s_j)]}{\mathbb{E}_{s_i, s_j \in S_{ano}, i \neq j}[D(s_i, s_j)]}$$

A larger NC indicates the abnormal points are closer to each other and increases the difficulty of anomaly detection.

Normalized adjacency of normal/abnormal cluster (NA) : NA is a measure that we propose and is defined as the ratio of the minimum distance between the centroids of normal and abnormal clusters to the average distance among the centroids of all normal clusters. Denote the set of centroids of normal clusters C_{nor} and the set of centroids of anomalous clusters C_{ano}, then

$$N_a = \frac{\min_{c_i \in C_{ano}, c_j \in C_{nor}} D(c_i, c_j)}{\mathbb{E}_{c_i, c_j \in C_{nor}, i \neq j}[D(c_i, c_j)]}$$

If the system only has one normal cluster, NA is null. A larger NA indicates abnormal points are more distant from the normal points.

RC is a general description of the clusteredness of the whole dataset, and NC is the clusteredness contrast between normal and anomalous data points. NC and NA require knowledge of the actual normal and abnormal subsequences and their clustering, so they can only be calculated on our synthetic datasets.

5 EXPERIMENTAL SETTINGS

In this section, we review the settings of the initial evaluation of 12 representative AD methods in TSB-UAD. The goal of this evaluation is *not* to be exhaustive, as such an effort would require the consideration of at least one hundred methods [13, 23, 49]. Instead, we focus on recent methods that are *representative* of the main AD categories (see Section 2) and have reported state-of-the-art performance in some of the datasets we have included in TSB-UAD. Variants of the included methods, or more sophisticated tuning and pre-processing steps for each method may lead to improved performance. We aim to show that TSB-UAD is a reliable and robust resource and we leave such exhaustive evaluation to future work.

Datasets: We use all 13766 time series included in TSB-UAD. Specifically, 1980 time series across the 18 public datasets, 958 time series across the 126 artificial datasets, and 10828 time series across 92 synthetic datasets. All time series points are annotated as normal or abnormal points to enable computation of evaluation measures.

Platform: We ran our experiments on a server with the the following configuration: Dual Intel(R) Xeon(R) Silver 4116 (12-core with 2-way SMT), 2.10 GHz, 196GB RAM. The server has an NVIDIA Quadro P6000 GPU and ran Ubuntu Linux 18.04.3 LTS (64-bit).

Implementation: We implemented the library and scripts that accompany TSB-UAD in Python 3.8 with the main following dependencies: sklearn 0.23.0, tensorflow 2.3.0, pandas 1.2.5, and networkx 2.6.3. For repeatability purposes and to ease experimentation we make datasets and code available: www.timeseries.org/TSB-UAD.

Algorithms: For the initial evaluation we consider the following strong baselines. Isolation Forest (IForest) [67] constructs the binary tree based on the space splitting and the nodes with shorter path lengths to the root are more likely to be anomalies. The Local Outlier Factor (LOF) [25] computes the ratio of the neighboring density to the local density. The Histogram-based Outlier Score (HBOS) [46] constructs a histogram for the data and the inverse of the height of the bin is used as the outlier score of the data point. Matrix Profile (MP) [116] calculates as anomaly the subsequence with the most significant 1-NN distance. NORMA [17] identifies the normal pattern based on clustering and calculates each point's effective distance to the normal pattern. Principal Component Analysis (PCA) [2] projects data to a lower-dimensional hyperplane, and data points with a significant distance from this plane can be identified as outliers. Autoencoder (AE) [97] projects data to the lower-dimensional latent space and reconstructs the data, and outliers are expected to have more evident reconstruction deviation. LSTM-AD [74], Polynomial Approximation (POLY) [63], and CNN [77] build a relationship between current and previous time series, and the outliers are detected by the deviation between the predicted and actual values. One-class Support Vector Machines (OCSVM) [98] fits the dataset to find the normal data's boundary.

Parameter Settings: For Isolation Forest we consider two variants: IForest1 indicates the IForest model without a sliding window, which is suited to the global point outliers, whereas IForest considers the sliding window variant. LSTM-AD, AE, CNN, and OCSVM are semi-supervised algorithms that require anomaly-free training data. In our test, only KDD21, NASA-SMAP, and NASA-MSL contain anomaly-free training data. For the other datasets, we train the models on the initial regions of the time series. Specifically, the training ratio for YAHOO is 30% and for the remaining datasets is 10%. We expect a low-density anomalies (< 5%) in the training dataset would not affect the result. However, we note that for some datasets with higher contamination ratios the results for the semi-supervised algorithms could likely further improved. We also highlight that several of the methods into consideration require minimal tuning and the default values reported in the corresponding papers or codes we rely on work well across datasets. For IForest/IForest1, we use the default 100 base estimators in the tree ensemble. For LOF, we follow the default setting in this model and we use 20 as the number of neighbors. For MP we set the window as the period of the time series, estimated using the autocorrelation function. We use the same period estimation for all methods requiring to set a a sliding window. For NORMA, we use NORMA-smpl [17] and we follow the default parameter settings in the paper. Similarly to MP, the pattern length is estimated with the autocorrelation function. We set the normal model of length to be 3*pattern length and sample 40% of the data without overlapping. For PCA, we use 10 principal components. For POLY, the best model is selected from the following settings. The power of polynomial fitted to the data is 0 or 3. The length of the window to be predicted is 20 or the period of the series. For OCSVM, we set the upper bound on the fraction of training errors to be 0.05. For LSTM-AD, we use the following parameters: two LSTM layers with units=50, then a Dense layer with units=1. loss='mse', optimizer='adam', validation split ratio=0.15, batch size=64, epochs = 50, patience = 5. For AE, the best model is selected from three MLP-based Autoencoders with the architectures given by (32,16,8,16,32), (32,8,32), (32,16,32). Activation function is ReLU. Each number indicates the units for the corresponding Dense layer. Then one Dense layer with units = the length of the input, validation split ratio=0.15, batch size=64, epochs=100, patience=5, optimizer='adam', loss='mse'. Finally, for CNN, we use three Convolutional Blocks (filters=8,16,32, kernel size=2, strides=1) with Max Pooling (pool size=2) and ReLU. Then one Dense layer with units=64, then one Dropout layer with rate=0.2, then one dense layer with units=1. loss='mse', optimizer='adam', validation split ratio=0.15, batch size=64, epochs = 100, patience = 5.

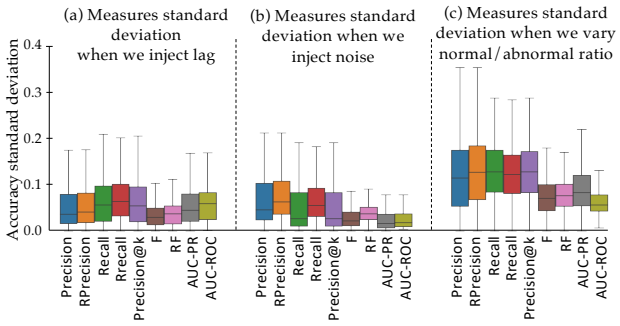


Figure 6: Box plots illustrate the standard deviation of the accuracy measures over the entire benchmark when we vary (a) lag; (b) noise; and (d) normal/abnormal ratio.

6 EXPERIMENTAL RESULTS

We first evaluate the robustness of the different accuracy measures (Section 6.1). Then, we compare the 12 methods in the 18 public datasets that are part of the TSB-UAD (Section 6.2 and Section 6.3). Along with the accuracy results we also present a runtime evaluation. For several of the methods with good performance, especially those without the need of supervision, we continue the evaluation in the artificial and the synthetic datasets. In Section 6.4, we concretely build the relation between the strength of transformation to the performance of algorithms based on the 92 synthetic datasets. Finally, in Section 6.5, we use NC, NA, and RC to quantify the difficulty of the artificial datasets and observe the correlation between these coefficients and the performance of the algorithms.

6.1 Evaluation of AD Accuracy Measures

We start by evaluating the different accuracy measures (Section 4.4) based on their sensitivity to the following factors: (i) noise on the anomaly score; (ii) lag on the anomaly score; and (iii) normal vs. abnormal ratio in the time series labels. The first two factors can be injected on the AD methods anomaly score due to manufacturing issues, external causes, or the AD method design. For instance, LOF and Isolation Forest, that both consider subsequences of a given length ℓ, might produce an anomaly score with a lag of ℓ with the labels. Thus, a strong accuracy evaluation measure requires to be as less influenced as possible to the previously enumerated factors.

For each AD method, we first compute the anomaly score S_T on a given time series. We then inject either lag l or noise n, or change the normal versus abnormal ratio r. For 10 different randomly selected values of $l \in [-0.25 * \ell, 0.25 * \ell]$, $n \in [-0.05 * (max(S_T) - min(S_T)), 0.05 * (max(S_T) - min(S_T))]$ and $r \in [0.01, 0.2]$, we compute the 9 different evaluation measures. For each evaluation measure, we compute the standard deviation of the ten different

values. We then compute the average standard deviation over all AD method. We thus obtain three values (average standard deviation when we vary the lag, noise, and normal versus abnormal ratio) for each accuracy measure and each time series in the benchmark. We finally compute box plots for lag (Figure 6(a)), for noise (Figure 6(b)), and for the normal/abnormal ratio (Figure 6(c)).

Our analysis confirms that AUC-ROC, F, and Range-F (RF) score are the most robust measures when we vary the noise and the normal versus abnormal ratio. Overall, AUC-ROC is more robust than F and RF for the noise and the normal/abnormal ratio, whereas F and RF are less sensitive to lag. Therefore, we focus on AUC-ROC that, in contrast to F and RF, does not require a threshold on the anomaly scores. Nonetheless, TSB-UAD computes all measures.

6.2 Benchmark Accuracy Evaluation

We first test 12 models on the 18 public datasets. Table 3 presents the average AUC-ROC (AUC) and F scores across all datasets and methods. Note that, due to threshold selection (for fairness we used the same threshold, Th, on the anomaly score S_T, $Th = \mu(S_T) + 3 * \sigma(S_T)$, for all methods), F values are 0 in a few cases, which verifies our choice of AUC for our analysis. Figure 7(a) depicts a boxplot per method corresponding to the AUC-ROC values over the entire benchmark. From this initial inspection, two methods seem to perform well: NORMA and MP.

To better understand the ranking of the methods, we use the Friedman test followed by Wilcoxon signed-rank test [113] in two levels of granularity: (i) fine-grained analysis that considers each time series separately; and (ii) aggregated analysis at the dataset level (using the average AUC-ROC of the time series in each dataset). Different levels of granularity provide better understanding and insights for the performance of the AD methods. From the fine-grained analysis we observe that NORMA and MP outperform the rest of the methods significantly (Figure 7(a.1)). However, the aggregated analysis shows that none of the methods are significantly better than the others (Figure 7(a.2)) and, importantly, MP is no longer among the highest ranked methods. We argue that both analyses are important. In the first case, we take into consideration 1980 different scenarios and cases with anomalies, which provides evidence for the statistical test to detect significant differences among AD methods. However, due to the imbalance of time series per dataset, methods that happen to work well on particular types of time series and anomalies that appear in the larger datasets might benefit from this analysis. The aggregated analysis provides a different angle, which may confirm or dispute the previous findings. We observe, that NORMA performs well in both analyses, ranked first in the fine-grained analysis and second in the aggragated analysis. POLY, AE, and IForest seem to also perform well in both scenarios.

We continue our analysis by comparing methods for two distinct sections of the benchmark: time series that contain point-based anomalies only (Figure 7(b)) and sequence-based anomalies only (Figure 7(c)). We first observe that the rankings of the methods are significantly different between these two categories. For point-based, CNN and LSTM are significantly better than the other methods. However, these two methods are the third and the second-worst for sequence-based anomalies. Overall, we observe that NORMA and MP perform well on these two categories (first and second for sequence-based and third and fourth for point-based).

Finally, we divide the two aforementioned categories into two more categories: time series that contain single (Figure 7(b.1) and

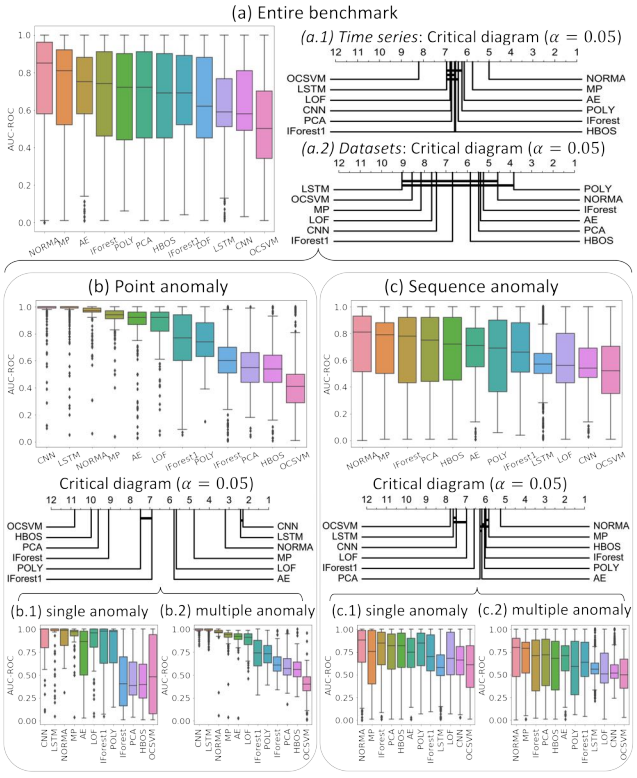


Figure 7: Accuracy evaluation of 12 AD methods over (a) the benchmark. (b) depicts the accuracy evaluation on time series that contain (b.1) single point anomaly and (b.2) multiple point anomalies. (c) depicts the evaluation on time series that contain (c.1) single collective anomaly and (c.2) multiple collective anomalies.

(c.1)) and multiple anomalies (Figure 7(b.2) and (c.2)). We do not observe big differences between these two categories. For instance, LSTM and CNN remain the best two methods for single and multiple point-based anomalies. Moreover, there were no significant differences between NORMA, POLY, IForest, HBOS, LOF, PCA, and MP for single sequence-based anomalies. However, the Wilcoxon signed-rank test demonstrates that NORMA outperforms MP and all other methods for multiple sequence-based anomalies significantly.

6.3 Use Cases: Comparisons Between Methods

Previously, we demonstrated that the performance of AD methods vary significantly among datasets and different anomaly types. In this section, we illustrate these variations with concrete examples.

Comparison between NORMA and MP: NORMA and MP are among the best methods (according to the fine-grained analysis above) and are designed to detect collective anomalies. In the ECG dataset, the average AUC for MP is significantly lower than NORMA. However, in the MGAB dataset, we observe the opposite trend: MP has a significantly better performance than NORMA. We plot the comparison between NORMA and MP in ECG and MGAB in Figure 8. NORMA dominates nearly all series in ECG while MP dominates all series in MGAB. This contrast is interesting because both ECG and MGAB data are periodic, with some collective outliers within one period that break the regular pattern. So we cannot predict which algorithm is the best by visually observing the data or types of anomalies. We need extra information: the cardinality of the abnormal points and the variation of the normal points.

Table 3: Average AUC and F accuracy measures for the 18 public datasets. Bold values indicate the best AUC result for each dataset.

Datasets	IForest		IForest1		LOF		MP		PCA		NORMA		HBOS		POLY		OCSVM		AE		CNN		LSTM	
	AUC	F	AUC	F	AUC	F	AUC	F	AUC	F	AUC	F	AUC	F	AUC	F	AUC	F	AUC	F	AUC	F	AUC	F
Dodgers	0.79	0.16	0.64	0.02	0.54	0.10	0.52	0.19	0.77	0.26	**0.79**	0.19	0.3	0.00	0.69	0.10	0.64	0.00	0.73	0.08	0.68	0.06	0.39	0.04
ECG	0.75	0.27	0.61	0.18	0.56	0.09	0.58	0.12	0.71	0.25	**0.95**	0.33	0.68	0.20	0.70	0.24	0.64	0.17	0.73	0.21	0.52	0.03	0.54	0.03
IOPS	0.54	0.04	**0.78**	0.25	0.50	0.08	0.72	0.10	0.74	0.17	0.76	0.12	0.64	0.11	0.68	0.11	0.71	0.14	0.63	0.13	0.61	0.09	0.61	0.10
MGAB	0.57	0.00	0.58	0.00	**0.96**	0.62	0.91	0.24	0.54	0.00	0.55	0.00	0.54	0.00	0.51	0.00	0.52	0.00	0.71	0.06	0.58	0.04	0.56	0.03
NAB	0.45	0.05	0.56	0.10	0.48	0.07	0.49	0.05	0.69	0.16	0.58	0.05	0.68	0.11	**0.75**	0.14	0.61	0.09	0.54	0.07	0.52	0.06	0.50	0.05
NASA-MSL	0.57	0.04	0.69	0.21	0.52	0.03	0.52	0.00	0.75	0.23	0.55	0.00	0.77	0.21	**0.81**	0.24	0.64	0.12	0.70	0.14	0.57	0.14	0.57	0.13
NASA-SMAP	0.72	0.21	0.68	0.13	0.68	0.20	0.62	0.18	0.74	0.25	0.80	0.19	0.77	0.27	**0.80**	0.20	0.65	0.31	0.77	0.33	0.68	0.17	0.64	0.14
SensorScope	0.56	0.00	0.56	0.06	0.55	0.09	0.50	0.02	0.54	0.05	0.59	0.01	0.56	0.02	**0.62**	0.09	0.51	0.04	0.52	0.03	0.52	0.04	0.53	0.05
YAHOO	0.62	0.06	0.81	0.18	0.86	0.11	0.86	0.06	0.57	0.06	0.92	0.11	0.57	0.07	0.76	0.08	0.50	0.03	0.79	0.06	**0.96**	0.47	0.94	0.46
KDD21	0.65	0.09	0.57	0.02	0.78	0.17	**0.90**	0.22	0.58	0.07	0.88	0.22	0.60	0.06	0.58	0.04	0.60	0.16	0.79	0.16	0.74	0.12	0.66	0.08
Daphnet	0.74	0.06	0.68	0.08	**0.78**	0.08	0.44	0.00	0.69	0.05	0.46	0.00	0.69	0.04	0.77	0.08	0.45	0.01	0.44	0.01	0.47	0.01	0.44	0.02
GHL	**0.94**	0.07	0.94	0.06	0.54	0.00	0.42	0.01	0.91	0.02	0.64	0.00	0.92	0.02	0.76	0.02	0.45	0.02	0.63	0.01	0.47	0.00	0.47	0.00
Genesis	0.78	0.00	0.66	0.19	0.68	0.00	0.35	0.00	0.85	0.00	0.6	0.00	0.59	0.00	**0.87**	0.32	0.70	0.03	0.72	0.01	0.73	0.02	0.53	0.01
MITDB	0.70	0.09	0.61	0.06	0.61	0.09	0.69	0.11	0.67	0.09	**0.86**	0.20	0.70	0.07	0.68	0.11	0.65	0.13	0.80	0.17	0.58	0.05	0.51	0.02
OPP	0.49	0.07	0.52	0.03	0.45	0.10	**0.82**	0.01	0.52	0.15	0.65	0.06	0.54	0.09	0.28	0.01	0.38	0.01	0.70	0.07	0.47	0.01	0.57	0.08
Occupancy	0.86	0.03	0.78	0.07	0.53	0.04	0.32	0.00	0.78	0.08	0.53	0.00	**0.89**	0.02	0.80	0.13	0.66	0.02	0.69	0.02	0.79	0.04	0.71	0.02
SMD	0.85	0.35	0.73	0.25	0.69	0.18	0.51	0.03	0.80	0.31	0.61	0.03	0.77	0.31	**0.87**	0.41	0.61	0.11	0.63	0.09	0.61	0.08	0.58	0.07
SVDB	0.72	0.19	0.58	0.08	0.59	0.14	0.74	0.17	0.68	0.19	**0.92**	0.33	0.71	0.15	0.67	0.18	0.68	0.15	0.79	0.18	0.58	0.07	0.55	0.06

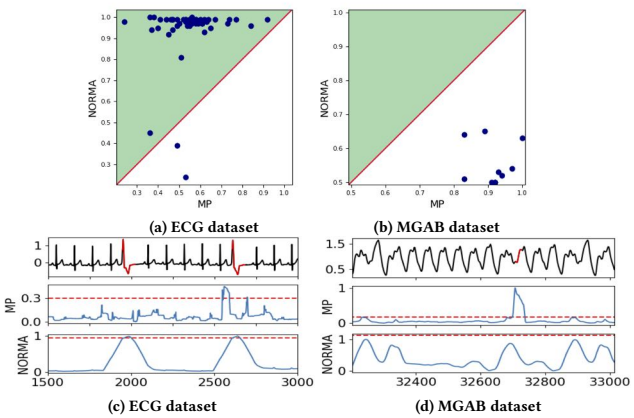


Figure 8: (a,b) The comparison of NORMA and MP over ECG and MGAB. NORMA outperforms MP in ECG and MP outperforms NORMA in MGAB. (c,d) The comparisons among anomaly scores for one ECG and one MGAB data. From top to bottom: Real data and MP and NORMA scores. ECG contains multiple repeated collective outliers and MGAB contains multiple normal patterns.

To better illustrate this point, Figure 8c shows a segment of the ECG data with its corresponding anomaly scores obtained from MP and NORMA. ECG data contains many repeated anomaly subsequences. MP calculates the 1-NN distance among subsequences but the distances for the abnormal points are also small. The impact of the repeated anomaly subsequences is reflected in the anomaly score. We observe a low anomaly score for MP at the center region of anomalies. In contrast, NORMA offers a good detection because NORMA does not consider the absolute distance to its neighboring points and is immune to the cardinality of the abnormal points.

Figure 8d shows a similar comparison for a MGAB time series. The critical difference between the MGAB and ECG datasets is that MGAB data contains several patterns while ECG data only has one pattern. Another difference is that the MGAB has only 10 collective outliers per series while ECG has 195.6 collective outliers per series. The collective outliers in MGAB are sparse and quite different from each other. In this scenario, MP has a high performance to detect the isolated anomaly. NORMA calculates the weighted distance of each point to all normal centroids. The variation of the normal clusters makes normal points to also get a large distance (Figure 8d), which complicates the detection of anomalies for NORMA in that case.

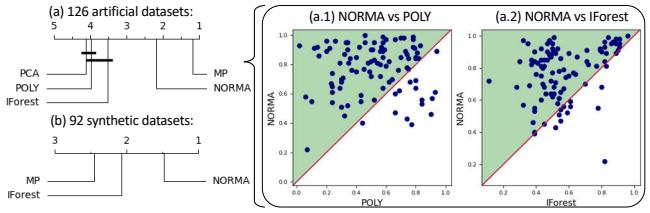


Figure 9: Ranking of algorithms based on the average of their ranks across (a) 126 artificial datasets and (b) 92 synthetic datasets. $\alpha = 0.05$. (a.1,a.2) Comparison of NORMA, POLY, and IForest over 126 artifical datasets. Each point in the scatter plot indicates one dataset.

The comparison among these two algorithms show the importance of two factors: number of anomalies and anomaly normality.

Comparisons among NORMA, MP, IForest, and POLY: We also apply Nemenyi test on 126 artificial datasets (Fig. 9(a)) and 92 synthetic datasets (Fig. 9(b)), respectively. Interestingly, MP is the top measure in the artificial datasets with an average rank of 1.18 and is significantly better than NORMA with a 95% confidence level. In contrast, in the synthetic datasets, NORMA is the top measure with an average rank of 1.48 and significantly outperforms MP with a 95% confidence level. This contradictory result comes from an artificial bias rooted in the anomaly construction method. When we construct the series in the artificial datasets, we only pick one or two subsequences from the abnormal labels and twenty subsequences from every normal label. Due to this unbalanced sampling, the average distance among normal subsequences is usually smaller than the one among abnormal subsequences. So statistically, the anomaly subsequence has a much larger 1-NN distance than the normal subsequences, and MP benefits from this property. We also observe that NORMA significantly outperforms IForest and that IForest and POLY do not present a significant difference. PCA is ranked last and does not present a significant difference from POLY. We also plot the AUC comparison over each dataset between NORMA and POLY (Fig. 9(a.1)). Most of the datasets fall above the diagonal line of the scatter plot, which forms a contrast with the good performance of POLY in the public datasets. A similar comparison between NORMA and IForest is shown in Fig. 9(a.2). NORMA outperforms IForest in 111 out of 126 datasets.

Next, we focus on three of the strongest methods (MP, NORMA, IForest) for the artificial datasets and re-test them in 92 synthetic datasets obtained by applying eight different global or local transformations with varying strengths of disorders or contamination ratios.

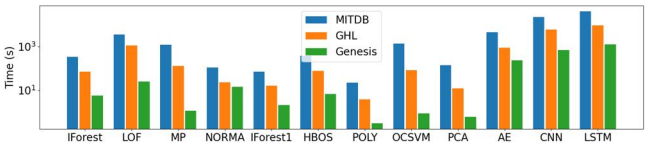


Figure 10: Computation time of different algorithms on MITDB, GHL, and Genesis dataset. Y-axis is in log10-scale. All computations have been performed on a single process in CPU.

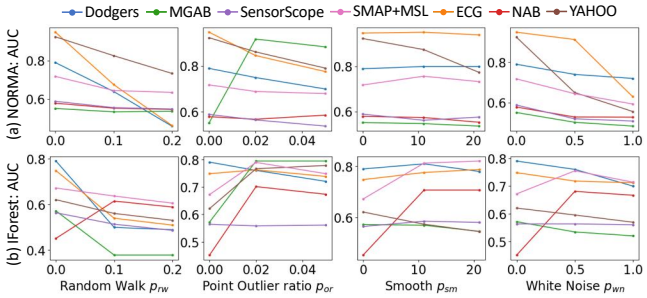


Figure 11: AUC vs global transformations for each dataset based on NORMA and IForest. x-axis indicates the strength.

We believe the Nemenyi test on this catogory is representative for the general case. Figure 9(b) shows that NORMA is significantly better than MP. In addition, NORMA significantly outperforms IForest, which is consistent with the result in the artificial datasets. Finally, Figure 10 shows the comparison of the computation time for all models based on three datasets (ran in CPU). The runtime of neural network models are an order of magnitude higher than the other models. POLY and PCA achieve the best runtime performance.

6.4 Impact of transformations

In this section, we discuss the effects of local and global transformations over the public datasets. For the transformations without introducing new outliers (adding random-walk background, smoothing, or add white noise), we build the relationship between the average AUC of NORMA and IForest and the strength of transformations in Fig. 11. We observe the negative correlation between the AUC and transformation strength for random-walk background and white noise, which is consistent with the intuition. There is no evident correlation between the AUC and smoothing window. The only exception occurs at the NAB dataset with IForest algorithm. In the first panel of Fig. 11(b), the AUC of the NAB dataset first increases from 0.452 to 0.615, then drops to 0.589. A similar pattern also occurs in the white noise panel (Fig. 11(b)).

Figure 12 shows the trend of average AUC over all synthetic datasets relative to the strengths of three anomaly-free transformations. NORMA outperforms IForest and MP under all cases. We observe a flipping of AUC between IForest and MP around white noise $p_{wn} = 0.25$, so IForest is more resilient compared to MP.

For the transformations that introduce new outliers, for example, adding point anomalies, adding some pattern-related, or frequency-related anomalies, we also build the relationship between the average AUC and the contamination ratio introduced by the corresponding transformations (Fig. 13). We do not observe any strong correlations as it is difficult to isolate new from old anomalies. However, new anomalies do not degrade the overall performance.

6.5 Difficulty of dataset

To supplement the Friedman test and describe the difficulty of the dataset from another perspective, we build the correlation between

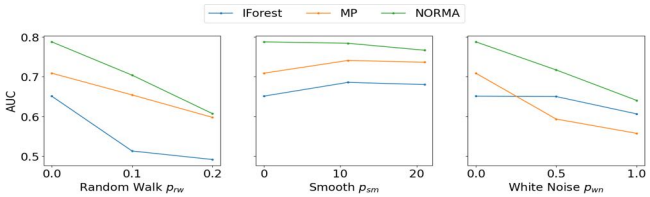


Figure 12: The variations of the average AUC relative to the strength of transformations over all datasets.

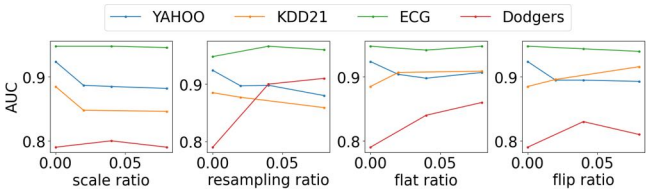


Figure 13: AUC vs. Local transformed data.x-axis indicates the contamination ratio of the local transformations.

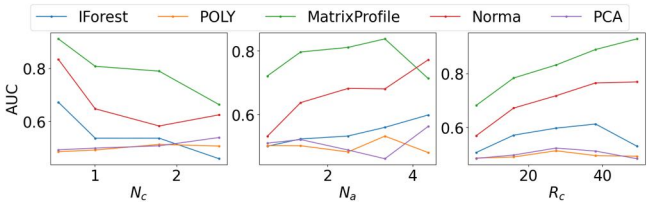


Figure 14: Avg. AUC vs. N_c, N_a, and R_c on synthetic datasets.

the average AUC with NC, NA, and RC, which describe the relative position between normal and abnormal data (Fig. 14). POLY and PCA's average AUC fluctuate around 0.5, so we focus on the other three models: MP, NORMA, and IForest. A larger NC indicates outliers are closer to each other and are more likely to form clusters. The AUC for the three algorithms decrease 27%, 25%, and 31%, respectively, with the increase of NC from 0.5 to 2.5. When NC>1, NORMA has a relatively stable performance compared with MP. MP depends on the 1-NN distance. Hence, the clustering of abnormal points has a more evident impact on MP. Similarly, it is also hard for IForest to isolate an anomaly point from a cluster. A larger NA indicates the outlier is more distant from the normal cluster. We observe a positive correlation between AUC and NA for IForest and NORMA, which increase 20% and 45% when NA increases from 0.6 to 4.3. MP does not present a strong correlation with NA. We also observe the improvement of NORMA in this case.

7 CONCLUSION

In this work, we proposed TSB-UAD an end-to-end benchmark for univariate time-series anomaly detection. Our benchmark covers a wide range of anomalies, provides eleven transformation methods to emulate the types of anomalies, and synthesizes datasets with different levels of similarity between normal and abnormal subsequences. We believe TSB-UAD can provide a uniform platform to compare different methods across different realistic scenarios and assist in the identification of robust AD methods.

ACKNOWLEDGMENTS

We thank the anonymous reviewers whose comments have greatly improved this manuscript. This research was supported in part by NetApp, Cisco Systems, Exelon Utilities, and HPC resources from GENCIIDRIS (Grants 2020-101471, 2021-101925).

REFERENCES

[1] [n.d.]. http://iops.ai/dataset_detail/?id=10.

[2] Charu C. Aggarwal. 2017. *Outlier Analysis* (2 ed.). Springer International Publishing. https://doi.org/10.1007/978-3-319-47578-3

[3] Subutai Ahmad, Alexander Lavin, Scott Purdy, and Zuha Agha. 2017. Unsupervised real-time anomaly detection for streaming data. *Neurocomputing* 262 (2017), 134–147. https://doi.org/10.1016/j.neucom.2017.04.070

[4] Arvind Arasu, Mitch Cherniack, Eduardo Galvez, David Maier, Anurag S Maskey, Esther Ryvkina, Michael Stonebraker, and Richard Tibbetts. 2004. Linear road: a stream data management benchmark. In *Proceedings of the Thirtieth international conference on Very large data bases-Volume 30*. 480–491.

[5] Marc Bachlin, Meir Plotnik, Daniel Roggen, Inbal Maidan, Jeffrey M. Hausdorff, Nir Giladi, and Gerhard Troster. 2010. Wearable Assistant for Parkinsons Disease Patients With the Freezing of Gait Symptom. *IEEE Transactions on Information Technology in Biomedicine* 14, 2 (2010), 436–446. https://doi.org/10.1109/TITB.2009.2036165

[6] Anthony Bagnall, Hoang Anh Dau, Jason Lines, Michael Flynn, James Large, Aaron Bostrom, Paul Southam, and Eamonn Keogh. 2018. The UEA multivariate time series classification archive, 2018. *arXiv preprint arXiv:1811.00075* (2018).

[7] Anthony Bagnall, Jason Lines, Aaron Bostrom, James Large, and Eamonn Keogh. 2017. The great time series classification bake off: a review and experimental evaluation of recent algorithmic advances. *Data mining and knowledge discovery* 31, 3 (2017), 606–660.

[8] Anthony J. Bagnall, Richard L. Cole, Themis Palpanas, and Konstantinos Zoumpatianos. 2019. Data Series Management (Dagstuhl Seminar 19282). *Dagstuhl Reports* 9, 7 (2019), 24–39. https://doi.org/10.4230/DagRep.9.7.24

[9] Andrei Barbu, David Mayo, Julian Alverio, William Luo, Christopher Wang, Danny Gutfreund, Joshua Tenenbaum, and Boris Katz. 2019. Objectnet: A large-scale bias-controlled dataset for pushing the limits of object recognition models. (2019).

[10] Mohini Bariya, Alexandra von Meier, John Paparrizos, and Michael J Franklin. 2021. k-ShapeStream: Probabilistic Streaming Clustering for Electric Grid Events. In *2021 IEEE Madrid PowerTech*. IEEE, 1–6.

[11] Pawel Benecki, Szymon Piechaczek, Daniel Kostrzewa, and Jakub Nalepa. 2021. Detecting Anomalies in Spacecraft Telemetry Using Evolutionary Thresholding and LSTMs. In *Proceedings of the Genetic and Evolutionary Computation Conference Companion* (Lille, France) *(GECCO '21)*. Association for Computing Machinery, New York, NY, USA, 143144. https://doi.org/10.1145/3449726.3459411

[12] Ana Maria Bianco, M Garcia Ben, EJ Martinez, and Vıctor J Yohai. 2001. Outlier detection in regression models with arima errors using robust estimates. *Journal of Forecasting* 20, 8 (2001), 565–579.

[13] Ane Blázquez-García, Angel Conde, Usue Mori, and Jose A Lozano. 2021. A Review on outlier/Anomaly Detection in Time Series Data. *ACM Computing Surveys (CSUR)* 54, 3 (2021), 1–33.

[14] Peter Bodik, Armando Fox, Michael J Franklin, Michael I Jordan, and David A Patterson. 2010. Characterizing, modeling, and generating workload spikes for stateful services. In *Proceedings of the 1st ACM symposium on Cloud computing*. 241–252.

[15] Paul Boniol, Michele Linardi, Federico Roncallo, and Themis Palpanas. 2020. Automated Anomaly Detection in Large Sequences. In *36th IEEE International Conference on Data Engineering, ICDE 2020, Dallas, TX, USA, April 20-24, 2020*.

[16] Paul Boniol, Michele Linardi, Federico Roncallo, and Themis Palpanas. 2020. SAD: An Unsupervised System for Subsequence Anomaly Detection. In *36th IEEE International Conference on Data Engineering, ICDE 2020, Dallas, TX, USA, April 20-24, 2020*. IEEE, 1778–1781. https://doi.org/10.1109/ICDE48307.2020.00168

[17] Paul Boniol, Michele Linardi, Federico Roncallo, Themis Palpanas, Mohammed Meftah, and Emmanuel Remy. 2021. Unsupervised and scalable subsequence anomaly detection in large data series. *The VLDB Journal* 30, 6 (2021).

[18] Paul Boniol and Themis Palpanas. 2020. Series2Graph: Graph-based Subsequence Anomaly Detection for Time Series. *PVLDB* 13, 11 (2020).

[19] Paul Boniol, Themis Palpanas, Mohammed Meftah, and Emmanuel Remy. 2020. GraphAn: Graph-based Subsequence Anomaly Detection. *Proc. VLDB Endow.* 13, 12 (2020), 2941–2944. https://doi.org/10.14778/3415478.3415514

[20] Paul Boniol, John Paparrizos, Themis Palpanas, and Michael J Franklin. 2021. SAND in action: subsequence anomaly detection for streams. *Proceedings of the VLDB Endowment* 14, 12 (2021), 2867–2870.

[21] Paul Boniol, John Paparrizos, Themis Palpanas, and Michael J Franklin. 2021. SAND: streaming subsequence anomaly detection. *Proceedings of the VLDB Endowment* 14, 10 (2021), 1717–1729.

[22] Loic Bontemps, James McDermott, Nhien-An Le-Khac, et al. 2016. Collective anomaly detection based on long short-term memory recurrent neural networks. In *International Conference on Future Data and Security Engineering*. Springer, 141–152.

[23] Mohammad Braei and Sebastian Wagner. 2020. Anomaly detection in univariate time-series: A survey on the state-of-the-art. *arXiv preprint arXiv:2004.00433* (2020).

[24] Markus M. Breunig, Hans-Peter Kriegel, Raymond T. Ng, and Jörg Sander. 2000. LOF: Identifying Density-based Local Outliers. In *SIGMOD*.

[25] Markus M. Breunig, Hans-Peter Kriegel, Raymond T. Ng, and Jörg Sander. 2000. LOF: identifying density-based local outliers. *ACM SIGMOD Record* 29, 2 (May 2000), 93–104. https://doi.org/10.1145/335191.335388

[26] Yingyi Bu, Oscar Tat-Wing Leung, Ada Wai-Chee Fu, Eamonn J. Keogh, Jian Pei, and Sam Meshkin. 2007. WAT: Finding Top-K Discords in Time Series Database. In *SIAM*.

[27] Luis M. Candanedo and Véronique Feldheim. 2016. Accurate occupancy detection of an office room from light, temperature, humidity and CO2 measurements using statistical learning models. *Energy and Buildings* 112 (2016), 28–39. https://doi.org/10.1016/j.enbuild.2015.11.071

[28] Raghavendra Chalapathy and Sanjay Chawla. 2019. Deep learning for anomaly detection: A survey. *arXiv preprint arXiv:1901.03407* (2019).

[29] Cody Coleman, Daniel Kang, Deepak Narayanan, Luigi Nardi, Tian Zhao, Jian Zhang, Peter Bailis, Kunle Olukotun, Chris Ré, and Matei Zaharia. 2019. Analysis of dawnbench, a time-to-accuracy machine learning performance benchmark. *ACM SIGOPS Operating Systems Review* 53, 1 (2019), 14–25.

[30] Brian F Cooper, Adam Silberstein, Erwin Tam, Raghu Ramakrishnan, and Russell Sears. 2010. Benchmarking cloud serving systems with YCSB. In *Proceedings of the 1st ACM symposium on Cloud computing*. 143–154.

[31] Hoang Anh Dau, Eamonn Keogh, Kaveh Kamgar, Chin-Chia Michael Yeh, Yan Zhu, Shaghayegh Gharghabi, Chotirat Ann Ratanamahatana, Yanping, Bing Hu, Nurjahan Begum, Anthony Bagnall, Abdullah Mueen, Gustavo Batista, and Hexagon-ML. 2018. The UCR Time Series Classification Archive. https://www.cs.ucr.edu/~eamonn/time_series_data_2018/.

[32] Janez Demšar. 2006. Statistical comparisons of classifiers over multiple data sets. *The Journal of Machine Learning Research* 7 (2006), 1–30.

[33] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. 2009. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*. Ieee, 248–255.

[34] Dheeru Dua, Casey Graff, et al. 2017. UCI machine learning repository. (2017).

[35] Adam Dziedzic, John Paparrizos, Sanjay Krishnan, Aaron Elmore, and Michael Franklin. 2019. Band-limited training and inference for convolutional neural networks. In *International Conference on Machine Learning*. PMLR, 1745–1754.

[36] Andrew F. Emmott, Shubhomoy Das, Thomas Dietterich, Alan Fern, and Weng-Keen Wong. 2013. Systematic construction of anomaly detection benchmarks from real data. In *Proceedings of the ACM SIGKDD Workshop on Outlier Detection and Description (ODD '13)*. Association for Computing Machinery, New York, NY, USA, 16–21. https://doi.org/10.1145/2500853.2500858

[37] Hassan Ismail Fawaz, Germain Forestier, Jonathan Weber, Lhassane Idoumghar, and Pierre-Alain Muller. 2019. Deep learning for time series classification: a review. *Data mining and knowledge discovery* 33, 4 (2019), 917–963.

[38] Hassan Ismail Fawaz, Benjamin Lucas, Germain Forestier, Charlotte Pelletier, Daniel F Schmidt, Jonathan Weber, Geoffrey I Webb, Lhassane Idoumghar, Pierre-Alain Muller, and François Petitjean. 2020. Inceptiontime: Finding alexnet for time series classification. *Data Mining and Knowledge Discovery* 34, 6 (2020), 1936–1962.

[39] Pavel Filonov, Andrey Lavrentyev, and Artem Vorontsov. 2016. Multivariate Industrial Time Series with Cyber-Attack Simulation: Fault Detection Using an LSTM-based Predictive Data Model. arXiv:1612.06676 [cs.LG]

[40] Vincent Fortuin, Matthias Hüser, Francesco Locatello, Heiko Strathmann, and Gunnar Rätsch. 2018. Som-vae: Interpretable discrete representation learning on time series. *arXiv preprint arXiv:1806.02199* (2018).

[41] Anthony J Fox. 1972. Outliers in time series. *Journal of the Royal Statistical Society: Series B (Methodological)* 34, 3 (1972), 350–363.

[42] Milton Friedman. 1937. The use of ranks to avoid the assumption of normality implicit in the analysis of variance. *J. Amer. Statist. Assoc.* 32 (1937), 675–701.

[43] Ada Wai-Chee Fu, Oscar Tat-Wing Leung, Eamonn J. Keogh, and Jessica Lin. 2006. Finding Time Series Discords Based on Haar Transform. In *ADMA*.

[44] Sam George. 2019 (accessed August 15, 2020). *IoT Signals report: IoT's promise will be unlocked by addressing skills shortage, complexity and security.* https://blogs.microsoft.com/blog/2019/07/30/.

[45] Ahmad Ghazal, Tilmann Rabl, Minqing Hu, Francois Raab, Meikel Poess, Alain Crolotte, and Hans-Arno Jacobsen. 2013. Bigbench: Towards an industry standard benchmark for big data analytics. In *Proceedings of the 2013 ACM SIGMOD international conference on Management of data*. 1197–1208.

[46] Markus Goldstein and Andreas Dengel. 2012. Histogram-based outlier score (hbos): A fast unsupervised anomaly detection algorithm. *KI-2012: poster and demo track* 9 (2012).

[47] Jim Gray. 1993. The benchmark handbook for database and transasction systems. *Mergan Kaufmann, San Mateo* (1993).

[48] Scott David Greenwald. 1990. *Improved detection and classification of arrhythmias in noise-corrupted electrocardiograms using contextual information.* Thesis. Massachusetts Institute of Technology. https://dspace.mit.edu/handle/1721.1/29206 Accepted: 2005-10-07T20:45:22Z.

[49] Manish Gupta, Jing Gao, Charu C Aggarwal, and Jiawei Han. 2013. Outlier detection for temporal data: A survey. *IEEE Transactions on Knowledge and data Engineering* 26, 9 (2013), 2250–2267.

[50] Junfeng He, Sanjiv Kumar, and Shih-Fu Chang. 2012. On the difficulty of nearest neighbor search. In *Proceedings of the 29th International Coference on International Conference on Machine Learning (ICML'12)*. Omnipress, Madison, WI, USA, 41–48.

[51] Mark Hung. 2017. Leading the iot, gartner insights on how to lead in a connected world. *Gartner Research* (2017), 1–29.

[52] Alexander Ihler, Jon Hutchins, and Padhraic Smyth. 2006. Adaptive Event Detection with Time-Varying Poisson Processes. In *Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (Philadelphia, PA, USA) *(KDD '06)*. Association for Computing Machinery, New York, NY, USA, 207216. https://doi.org/10.1145/1150402.1150428

[53] Vincent Jacob, Fei Song, Arnaud Stiegler, Bijan Rad, Yanlei Diao, and Nesime Tatbul. 2020. Exathlon: A Benchmark for Explainable Anomaly Detection over Time Series. *arXiv preprint arXiv:2010.05073* (2020).

[54] Hoyoung Jeung, Sofiane Sarni, Ioannis Paparrizos, Saket Sathe, Karl Aberer, Nicholas Dawes, Thanasis G Papaioannou, and Michael Lehning. 2010. Effective metadata management in federated sensor networks. In *2010 IEEE International Conference on Sensor Networks, Ubiquitous, and Trustworthy Computing*. IEEE, 107–114.

[55] Hao Jiang, Chunwei Liu, Qi Jin, John Paparrizos, and Aaron J Elmore. 2020. Pids: attribute decomposition for improved compression and query performance in columnar storage. *Proceedings of the VLDB Endowment* 13, 6 (2020), 925–938.

[56] Hao Jiang, Chunwei Liu, John Paparrizos, Andrew A Chien, Jihong Ma, and Aaron J Elmore. 2021. Good to the Last Bit: Data-Driven Encoding with CodecDB. In *Proceedings of the 2021 International Conference on Management of Data*. 843–856.

[57] E. Keogh, T. Dutta Roy, U. Naik, and A Agrawal. [n.d.]. Multi-dataset Time-Series Anomaly Detection Competition 2021, https://compete.hexagon-ml.com/practice/competition/39/.

[58] Eamonn Keogh, Stefano Lonardi, Chotirat Ann Ratanamahatana, Li Wei, Sang-Hee Lee, and John Handley. 2007. Compression-based data mining of sequential data. *Data Mining and Knowledge Discovery* (2007).

[59] M. Kontaki, A. Gounaris, A. N. Papadopoulos, K. Tsichlas, and Y. Manolopoulos. 2011. Continuous monitoring of distance-based outliers over data streams. In *2011 IEEE 27th International Conference on Data Engineering*. 135–146. https://doi.org/10.1109/ICDE.2011.5767923

[60] Kwei-Herng Lai, Daochen Zha, Junjie Xu, Yue Zhao, Guanchu Wang, and Xia Hu. 2021. Revisiting Time Series Outlier Detection: Definitions and Benchmarks. In *NeurIPS Track on Datasets and Benchmarks*.

[61] N. Laptev, S. Amizadeh, and Y. Billawala. 2015. *S5 - A Labeled Anomaly Detection Dataset, version 1.0(16M)*. https://webscope.sandbox.yahoo.com/catalog.php?datatype=s&did=70

[62] Tae Jun Lee, Justin Gottschlich, Nesime Tatbul, Eric Metcalf, and Stan Zdonik. 2018. Greenhouse: A zero-positive machine learning system for time-series anomaly detection. *arXiv preprint arXiv:1801.03168* (2018).

[63] Zhi Li, Hong Ma, and Yongbing Mei. 2007. A Unifying Method for Outlier and Change Detection from Data Streams Based on Local Polynomial Fitting. In *Advances in Knowledge Discovery and Data Mining (Lecture Notes in Computer Science)*, Zhi-Hua Zhou, Hang Li, and Qiang Yang (Eds.). Springer, Berlin, Heidelberg, 150–161. https://doi.org/10.1007/978-3-540-71701-0_17

[64] Michele Linardi, Yan Zhu, Themis Palpanas, and Eamonn J. Keogh. 2020. Matrix Profile Goes MAD: Variable-Length Motif And Discord Discovery in Data Series. In *DAMI*.

[65] Chunwei Liu, Hao Jiang, John Paparrizos, and Aaron J Elmore. 2021. Decomposed bounded floats for fast compression and queries. *Proceedings of the VLDB Endowment* 14, 11 (2021), 2586–2598.

[66] Fei Tony Liu, Kai Ming Ting, and Zhi-Hua Zhou. 2008. Isolation Forest. In *ICDM (ICDM)*.

[67] Fei Tony Liu, Kai Ming Ting, and Zhi-Hua Zhou. 2008. Isolation Forest. In *2008 Eighth IEEE International Conference on Data Mining*. 413–422. https://doi.org/10.1109/ICDM.2008.17 ISSN: 2374-8486.

[68] Yubao Liu, Xiuwei Chen, and Fei Wang. 2009. Efficient Detection of Discords for Time Series Stream. *Advances in Data and Web Management* (2009).

[69] Haoran Ma, Benyamin Ghojogh, Maria N. Samad, Dongyu Zheng, and Mark Crowley. 2020. Isolation Mondrian Forest for Batch and Online Anomaly Detection. arXiv:2003.03692 [cs.LG]

[70] Spyros Makridakis and Michele Hibon. 2000. The M3-Competition: results, conclusions and implications. *International journal of forecasting* 16, 4 (2000), 451–476.

[71] Spyros Makridakis and Evangelos Spiliotis. 2021. The M5 Competition and the Future of Human Expertise in Forecasting. *Foresight: The International Journal of Applied Forecasting* 60 (2021).

[72] Spyros Makridakis, Evangelos Spiliotis, and Vassilios Assimakopoulos. 2018. The M4 Competition: Results, findings, conclusion and way forward. *International Journal of Forecasting* 34, 4 (2018), 802–808.

[73] Pankaj Malhotra, Lovekesh Vig, Gautam Shroff, and Puneet Agarwal. 2015. Long Short Term Memory Networks for Anomaly Detection in Time Series. (2015).

[74] Pankaj Malhotra, L. Vig, Gautam M. Shroff, and Puneet Agarwal. 2015. Long Short Term Memory Networks for Anomaly Detection in Time Series. In *ESANN*.

[75] G.B. Moody and R.G. Mark. 2001. The impact of the MIT-BIH Arrhythmia Database. *IEEE Engineering in Medicine and Biology Magazine* 20, 3 (2001), 45–50. https://doi.org/10.1109/51.932724

[76] George B Moody and Roger G Mark. 1992. MIT-BIH Arrhythmia Database. https://doi.org/10.13026/C2F305

[77] M. Munir, S. A. Siddiqui, A. Dengel, and S. Ahmed. 2019. DeepAnT: A Deep Learning Approach for Unsupervised Anomaly Detection in Time Series. *IEEE Access* 7 (2019), 1991–2005. https://doi.org/10.1109/ACCESS.2018.2886457

[78] Raghunath Othayoth Nambiar, Matthew Lanken, Nicholas Wakou, Forrest Carman, and Michael Majdalany. 2009. Transaction Processing Performance Council (TPC): twenty years later–a look back, a look ahead. In *Technology Conference on Performance Evaluation and Benchmarking*. Springer, 1–10.

[79] Peter Nemenyi. 1963. *Distribution-free Multiple Comparisons*. Ph.D. Dissertation. Princeton University.

[80] Irene CL Ng and Susan YL Wakenshaw. 2017. The Internet-of-Things: Review and research directions. *International Journal of Research in Marketing* 34, 1 (2017), 3–21.

[81] ES Page. 1957. On problems in which a change in a parameter occurs at an unknown point. *Biometrika* 44, 1/2 (1957), 248–252.

[82] Themis Palpanas and Volker Beckmann. 2019. Report on the First and Second Interdisciplinary Time Series Analysis Workshop (ITISA). *SIGMOD Rec.* 48, 3 (2019), 36–40. https://doi.org/10.1145/3377391.3377400

[83] Spiros Papadimitriou, Hiroyuki Kitagawa, Phillip B Gibbons, and Christos Faloutsos. 2003. Loci: Fast outlier detection using the local correlation integral. In *Proceedings 19th international conference on data engineering (Cat. No. 03CH37405)*. IEEE, 315–326.

[84] Ioannis Paparrizos. 2018. *Fast, Scalable, and Accurate Algorithms for Time-Series Analysis*. Ph.D. Dissertation. Columbia University.

[85] John Paparrizos. 2018. ucr time-series archive: Backward compatibility, missing values, and varying lengths. *URL: https://github. com/johnpaparrizos/UCRArchiveFixes* (2018).

[86] John Paparrizos, Paul Boniol, Themis Palpanas, Ruey S. Tsay, Aaron Elmore, and Michael J Franklin. 2022. Volume Under the Surface: A New Accuracy Evaluation Measure for Time-Series Anomaly Detection. *Technical Report LIPADE-TR-N7, Université Paris Cité* (2022).

[87] John Paparrizos and Michael J Franklin. 2019. Grail: efficient time-series representation learning. *Proceedings of the VLDB Endowment* 12, 11 (2019), 1762–1777.

[88] John Paparrizos and Luis Gravano. 2015. k-shape: Efficient and accurate clustering of time series. In *Proceedings of the 2015 ACM SIGMOD international conference on management of data*. 1855–1870.

[89] John Paparrizos and Luis Gravano. 2017. Fast and accurate time-series clustering. *ACM Transactions on Database Systems (TODS)* 42, 2 (2017), 1–49.

[90] John Paparrizos, Chunwei Liu, Bruno Barbarioli, Johnny Hwang, Ikraduya Edian, Aaron J Elmore, Michael J Franklin, and Sanjay Krishnan. 2021. VergeDB: A Database for IoT Analytics on Edge Devices.. In *CIDR*.

[91] John Paparrizos, Chunwei Liu, Aaron J Elmore, and Michael J Franklin. 2020. Debunking four long-standing misconceptions of time-series distance measures. In *Proceedings of the 2020 ACM SIGMOD International Conference on Management of Data*. 1887–1905.

[92] Charlotte Pelletier, Geoffrey I Webb, and François Petitjean. 2019. Temporal convolutional neural network for the classification of satellite image time series. *Remote Sensing* 11, 5 (2019), 523.

[93] Daniel Peña and Ruey S Tsay. 2021. *Statistical Learning for Big Dependent Data*. John Wiley & Sons.

[94] Tilmann Rabl, Christoph Brücke, Philipp Härtling, Stella Stars, Rodrigo Escobar Palacios, Hamesh Patel, Satyam Srivastava, Christoph Boden, Jens Meiners, and Sebastian Schelter. 2019. ADABench-Towards an Industry Standard Benchmark for Advanced Analytics. In *Technology Conference on Performance Evaluation and Benchmarking*. Springer, 47–63.

[95] Daniel Roggen, Alberto Calatroni, Mirco Rossi, Thomas Holleczek, Kilian Förster, Gerhard Tröster, Paul Lukowicz, David Bannach, Gerald Pirkl, Alois Ferscha, Jakob Doppler, Clemens Holzmann, Marc Kurz, Gerald Holl, Ricardo Chavarriaga, Hesam Sagha, Hamidreza Bayati, Marco Creatura, and José del R. Millàn. 2010. Collecting complex activity datasets in highly rich networked sensor environments. In *2010 Seventh International Conference on Networked Sensing Systems (INSS)*. 233–240. https://doi.org/10.1109/INSS.2010.5573462

[96] Mayu Sakurada and Takehisa Yairi. 2014. Anomaly detection using autoencoders with nonlinear dimensionality reduction. In *Proceedings of the MLSDA 2014 2nd workshop on machine learning for sensory data analysis*. 4–11.

[97] Mayu Sakurada and Takehisa Yairi. 2014. Anomaly Detection Using Autoencoders with Nonlinear Dimensionality Reduction. In *Proceedings of the MLSDA 2014 2nd Workshop on Machine Learning for Sensory Data Analysis* (Gold Coast, Australia QLD, Australia) *(MLSDA'14)*. Association for Computing Machinery, New York, NY, USA, 411. https://doi.org/10.1145/2689746.2689747

[98] Bernhard Schölkopf, Robert Williamson, Alex Smola, John Shawe-Taylor, and John Platt. 1999. Support vector method for novelty detection. In *Proceedings of the 12th International Conference on Neural Information Processing Systems (NIPS'99)*. MIT Press, Cambridge, MA, USA, 582–588.

[99] Pavel Senin, Jessica Lin, Xing Wang, Tim Oates, Sunil Gandhi, Arnold P. Boedihardjo, Crystal Chen, and Susan Frankenstein. 2015. Time series anomaly discovery with grammar-based compression. In *EDBT*.

[100] Nidhi Singh and Craig Olinsky. 2017. Demystifying Numenta anomaly benchmark. In *2017 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 1570–1577.

[101] Ya Su, Youjian Zhao, Chenhao Niu, Rong Liu, Wei Sun, and Dan Pei. 2019. Robust Anomaly Detection for Multivariate Time Series through Stochastic Recurrent Neural Network. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (Anchorage, AK, USA) *(KDD '19)*. Association for Computing Machinery, New York, NY, USA, 28282837.

https://doi.org/10.1145/3292500.3330672

[102] Sharmila Subramaniam, Themis Palpanas, Dimitris Papadopoulos, Vana Kalogeraki, and Dimitrios Gunopulos. 2006. Online Outlier Detection in Sensor Data Using Non-Parametric Models. In *VLDB 2006*. 187–198.

[103] Nesime Tatbul, Tae Jun Lee, Stan Zdonik, Mejbah Alam, and Justin Gottschlich. 2018. Precision and recall for time series. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*. 1924–1934.

[104] Nesime Tatbul, Tae Jun Lee, Stan Zdonik, Mejbah Alam, and Justin Gottschlich. 2018. Precision and Recall for Time Series. In *Advances in Neural Information Processing Systems*, Vol. 31. Curran Associates, Inc. https://proceedings.neurips.cc/paper/2018/hash/8f468c873a32bb0619eaeb2050ba45d1-Abstract.html

[105] Markus Thill, Wolfgang Konen, and Thomas Bäck. 2020. *MarkusThill/MGAB: The Mackey-Glass Anomaly Benchmark, https:// doi.org/ 10.5281/ zenodo.3762385*. https://doi.org/10.5281/zenodo.3762385

[106] Luan Tran, Liyue Fan, and Cyrus Shahabi. 2016. Distance-Based Outlier Detection in Data Streams. *Proc. VLDB Endow.* 9, 12 (Aug. 2016), 10891100.

[107] Ruey S Tsay. 1988. Outliers, level shifts, and variance changes in time series. *Journal of forecasting* 7, 1 (1988), 1–20.

[108] Ruey S Tsay and Rong Chen. 2018. *Nonlinear time series analysis*. Vol. 891. John Wiley & Sons.

[109] Ruey S Tsay, Daniel Pena, and Alan E Pankratz. 2000. Outliers in multivariate time series. *Biometrika* 87, 4 (2000), 789–804.

[110] Alexander von Birgelen and Oliver Niggemann. 2018. *Anomaly Detection and Localization for Cyber-Physical Production Systems with Self-Organizing Maps*. Springer Berlin Heidelberg, Berlin, Heidelberg, 55–71. https://doi.org/10.1007/978-3-662-57805-6_4

[111] Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R Bowman. 2018. GLUE: A multi-task benchmark and analysis platform for natural language understanding. *arXiv preprint arXiv:1804.07461* (2018).

[112] Frank Wilcoxon. 1945. Individual comparisons by ranking methods. *Biometrics Bulletin* (1945), 80–83.

[113] Frank Wilcoxon. 1945. Individual Comparisons by Ranking Methods. *Biometrics Bulletin* 1, 6 (1945), 80–83. http://www.jstor.org/stable/3001968

[114] Renjie Wu and Eamonn J Keogh. 2020. Current Time Series Anomaly Detection Benchmarks are Flawed and are Creating the Illusion of Progress. *arXiv preprint arXiv:2009.13807* (2020).

[115] Yuan Yao, Abhishek Sharma, Leana Golubchik, and Ramesh Govindan. 2010. Online anomaly detection for sensor systems: A simple and efficient approach. *Performance Evaluation* 67, 11 (2010), 1059–1075. https://doi.org/10.1016/j.peva.2010.08.018 Performance 2010.

[116] Chin-Chia Michael Yeh, Yan Zhu, Liudmila Ulanova, Nurjahan Begum, Yifei Ding, Hoang Anh Dau, Zachary Zimmerman, Diego Furtado Silva, Abdullah Mueen, and Eamonn Keogh. 2018. Time series joins, motifs, discords and shapelets: a unifying view that exploits the matrix profile. *Data Mining and Knowledge Discovery* 32, 1 (Jan. 2018), 83–123. https://doi.org/10.1007/s10618-017-0519-9

[117] Chin-Chia Michael Yeh, Yan Zhu, Liudmila Ulanova, Nurjahan Begum, Yifei Ding, Hoang Anh Dau, Diego Furtado Silva, Abdullah Mueen, and Eamonn J. Keogh. 2016. Matrix Profile I: All Pairs Similarity Joins for Time Series. In *ICDM*.

[118] Chuxu Zhang, Dongjin Song, Yuncong Chen, Xinyang Feng, Cristian Lumezanu, Wei Cheng, Jingchao Ni, Bo Zong, Haifeng Chen, and Nitesh V Chawla. 2019. A deep neural network for unsupervised anomaly detection and diagnosis in multivariate time series data. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 33. 1409–1416.